4.9 Systems Management Subsystem Overview

The Systems Management Subsystem (MSS) provides a complement of tools and services to manage ECS operations. The management services provided cover five major areas including fault, configuration, accountability, performance, and security (FCAPS). The MSS is implemented using COTS products customized to meet ECS requirements. The MSS maintains policy neutrality in implementing ECS management support.

The MSS software is installed at the Local System Management (LSM) position of each DAAC to manage production operations. The MSS software is also installed at the System Management Center (SMC) at GSFC to monitor and coordinate activities involving multiple sites and to perform designated common support functions for all sites.

Systems Management Subsystem Context

Figure 4.9-1 is the System Management Subsystem context diagram. The external systems referred to in the context diagram are EDOS, ASTER, NSI, Version 0 (V0) Information Management System (IMS), ESDIS, Science Users, and the Landsat 7 LPGS. Table 4.9-1 provides descriptions of the interface events shown in the MSS context diagram.

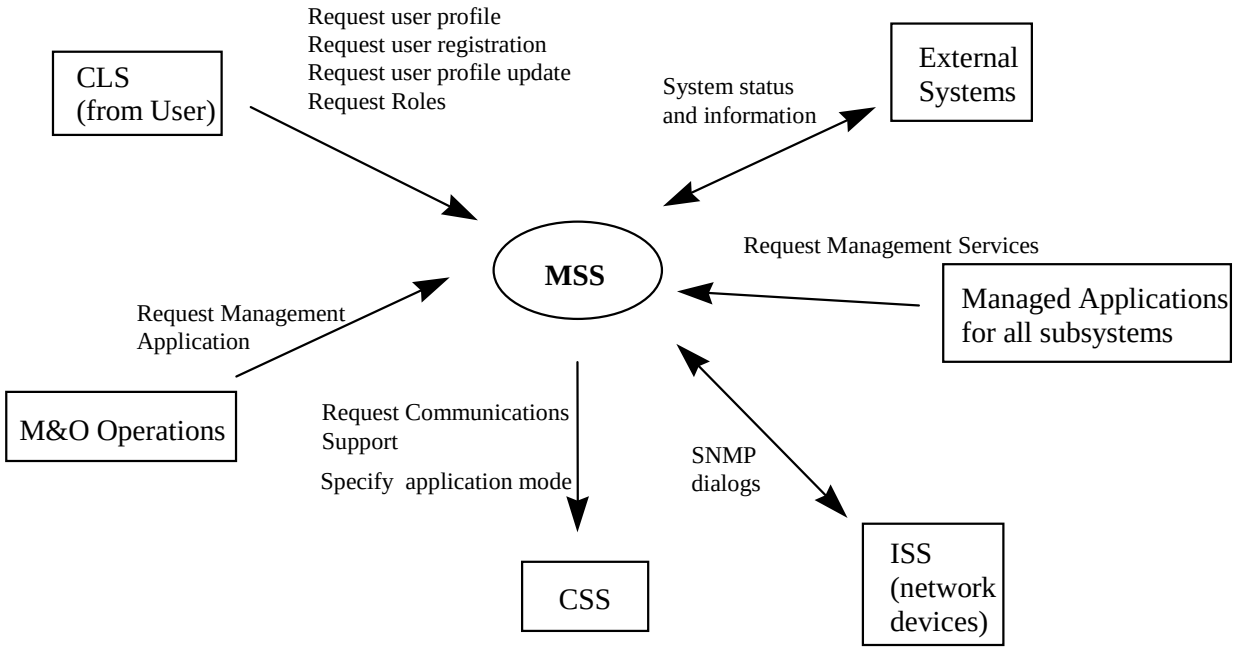

Figure 4.9-1. System Management Subsystem Context Diagram

Table 4.9-1. System Management Subsystem Interface Events (1 of 2)

Event	Interface Event Description
Request management application	Maintenance and Operations (M&O) staff interact with the MSS management application service tools in the fault, configuration, accountability, performance and security management areas. These tools enable the Operations staff to collect information/metrics, schedule resources for maintenance, monitor and analyze trends, maintain the base-line and schedules, and maintain user profiles
Request user registration	The unregistered ECS "guest" user submits a registration request to become a registered ECS user through the CLS that passes the request to the MSS. An M&O administrator processes the request and returns an approval status.
Request user profile update	A registered user can update user profile parameters such as e-mail address and mailing address through the CLS interface that passes the update to the MSS. The MSS updates the User Profile in the MSS database with the parameters.
Request roles	An M&O staff member is given a range of management roles to serve. The CLS sends a request to the MSS to identify the roles an M&O user can play so the MSS can bring up the corresponding software tool icons on the Operations staff member's desktop display.
Request management services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On start-up, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run. The MSS also interfaces with other subsystems to perform the following: • **DMS Order/Request tracking update** - The DMS interfaces with the MSS Order/Request Tracking service to create a user product order. • **User Profile Request** - The MSS provides requesting subsystems with access to User Profile parameters such as e-mail address and shipping address to support their processing activities.
Simple Network Management Protocol (SNMP) dialogs	The MSS monitors and controls network devices such as routers and concentrators using the industry standard SNMP protocol.
Request User Profile	This a request to MSS from other ECS subsystems or external systems (e.g., V0 IMS) to retrieve user profile information such as mailing, billing and shipping addresses, phone number, and electronic mail address.
System status and information	The MSS exchanges system status, trouble reports, and management report information with external systems such as the ASTER GDS, NSI V0 IMS, and the Landsat 7 LPGS via the EBnet.

Table 4.9-1. System Management Subsystem Interface Events (2 of 2)

Event	Interface Event Description
Request communications support	The CSS provides a library of services available to each ECS subsystem. The services required to perform specific subsystem assignments are requested by the subsystem from the CSS. These services include: DCE support, file transfer services, Network and Distributed File Servers, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services and Mode information.
Specify Application Mode	Prior to the start-up of any managed applications within a subsystem, the MSS provides the mode of operation to the CSS. Managed applications use the application interface PFGETMODE to obtain the operational mode to determine the servers to service the managed applications requests.

Systems Management Subsystem Structure

The MSS is comprised of three CSCIs and one hardware CI as follows:

The Management Software CSCI (MCI) provides distributed system management support capabilities in the fault, configuration, accountability, performance, and security service areas. Its Computer Software Components (CSCs) include:

- Network and Enterprise Management Framework: This CSC enables M&O to monitor and control communications devices, hosts, and applications in the distributed system. It also provides the framework for integrating a range of other management service applications.
- Security: The security service is implemented using a variety of free-ware or public domain packages which monitor and evaluate the various aspects of the security setup at each DAAC and reports status.
- Accountability Management: The accountability management support is provided by custom developed software for user registration and user profile attribute updates. The accountability management CSC also provides a tracking mechanism for user product orders.
- Trouble Ticket: The Trouble Ticket CSC manages system problem reports submitted by users and by external systems. The trouble ticket CSC also records problem assignees, tracks investigation progress, and provides users with problem resolution status.
- Network Backup/Restore: The Network backup and restore CSC enables the Operations staff to perform system backups and restores from a central administration position (at each DAAC).

- ASTER Standard Header Handler: The ASTER standard header handler CSC supports the ECS to ASTER GDS interface and inserts a standard header for e-mail messages sent to ASTER and removes the standard header from e-mail received from the ASTER GDS. The sequentially numbered messages are logged and can be resent by M&O staff for recovery from transfer problems.

The Management Agent CSCI (MACI) is the interface between each DAAC's master station and the ECS developed applications and COTS products distributed on hosts throughout the local area network. The MACI CSCI assures secure communications with ECS developed and COTS applications by using Remote Procedure Calls (RPCs) instead of the SNMP. The MACI CSCI implements the ECS application MIB that is a look-up table and contains configuration information about ECS applications. The ECS application MIB is queried by the Operations staff via the HP OpenView interface. The MACI CSCI accepts lifecycle commands from the master station to start-up/shutdown applications and forwards application status to the master station.

The Management Logistics CSCI (MLCI) supports the configuration management of the ECS. The MLCI CSCI is the following six CSCs:

- Baseline Manager (BLM): The BLM maintains records of the baseline operational configuration. The BLM CSC identifies hardware/software items, sites, versions, interdependencies, and tracks change history.

- Inventory, Logistics, Maintenance (ILM) Manager: The ILM CSC maintains records on contract purchased items containing information such as vendor, date of receipt, installation, warranty expiration, and licensing information. The ILM CSC also maintains maintenance records on contract purchased items.

- Software Change Manager: The software change manager CSC supports maintenance and change control of the science software configuration at each DAAC.

- Change Request Manager: The change request manager CSC tracks and maintains Configuration Change Requests (CCRs) at each DAAC.

- Software Distribution Manager: The software distribution manager CSC supports the distribution of ECS software, database information, software documentation, and COTS files to other various ECS destinations.

- Software License Manager: The software license manager CSC monitors and controls licensing of COTS products installed in the ECS.

The MSS hardware consists of a single hardware configuration item, the MHCI, provided at the SMC, the LSM positions at the Earth Observing System Operations Center (EOC), and each DAAC. The MHCI includes an enterprise monitoring server, a local management server, a management workstation, and printers. The MHCI provides processing and storage support for the execution of the management applications within the MCI, MLCI, and part of the MACI CSCIs. The MHCI provides a warm or cold standby for all MSS servers for fail-over capability.

Use of COTS in the Systems Management Subsystem

The MSS design uses COTS software to implement and provide management services as described below. Detailed explanations of the COTS software are provided in the CSC descriptions.

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide strings and collections. These libraries must be installed with the MSS software for any of the MSS custom processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ class libraries interact with the Sybase database Structured Query Language (SQL) server. The use of DBTools buffers the MSS processes from the relational database used. These libraries must be installed with the MSS software for any of the MSS custom processes to run.

- ICS' Builder Xcessory

  The Builder Xcessory GUI builder tool modifies the displays of MSS GUIs. The Builder Xcessory tool also generates the C++ code that produces MSS GUIs at run time. No operational part of this tool is needed at run-time.

- Sybase (SQL Server)

  Sybase's SQL server provides access for MSS to insert, update, and delete MSS database information. The Sybase SQL Server must be running during operations for the User Profile Server, Order Tracking Server, and Management Data Access (MDA) Server to operate.

- Crack

  Crack is a security management program that identifies user passwords that can be easily guessed. Crack enables systems administrators to force users to create passwords that more difficult for a potential intruder to exploit.

- Anlpassword

  Anlpassword is a security management program that enables system administrators to set certain rules for password creation (e.g., must be at least 8 characters long and contain a number or symbol). The anlpassword program makes it more difficult for passwords to be guessed and exploited by potential intruders.

- TCP Wrappers

  TCP Wrappers is a security management program that monitors and controls user applications that connect to various network services, such as TFTP, EXEC, FTP, RSH, TELNET, RLOGIN, FINGER, and SYSTAT. The actions performed by the TCP Wrappers program are configurable, but consist of logging the remote host name and performing basic checks on the request origin.

- SATAN

  SATAN is a security management program that helps systems administrators identify common networking-related security problems and reporting the problems without actually exploiting them. For each type or problem found, SATAN offers a tutorial that explains the problem and what its impact could be. The tutorial also explains what can be done about the problem: correct an error in a configuration file, install a bug fix from the vendor, use other means to restrict access, or simply disable the service.

- Tripwire

  Tripwire is security management program that is an integrity monitor. Tripwire uses several checksum/signature routines to detect file changes and monitors selected items of system-maintained information. Tripwire monitors permission, link, file size, and directory changes. It also detects file additions or deletions based on selected directories that are watched.

- ClearCase

  ClearCase is a UNIX software change management application used to maintain algorithms at each DAAC.

- XRP II

  XRP II is a configuration management tool that performs two basic functions within ECS: baseline management and inventory, logistics, and maintenance management.

- ACCELL

  ACCELL is a relational database used by the XRP II products. ACCELL must be installed on the same machine as XRP II.

- Networker

  Networker is an application which provides capabilities to backup and restore files or directories for all ECS hosts. Networker provides an interface for the system administrator to identify the files or directories for back up or restoring and performs the backup or restore according to specifications.

- DDTS

  DDTS is a UNIX based configuration management tool to handle configuration change requests (CCRs) in the ECS system. DDTS provides the user the capability to generate, monitor, and report on ECS CCRs.

- Remedy's Action Request System (ARS)

  The Remedy ARS (usually referred to as "Remedy") is a trouble ticketing application. Remedy generates, monitors, and reports on trouble tickets within the ECS. A custom ECS web interface to Remedy provides registered users the capability to generate and obtain status of the ECS trouble tickets via the web. Remedy also provides the DAAC User Services operators with a User Contact Log to maintain records of all contacts with ECS end users.

- BMC's Optima (formerly Peer's) Master Agent

  The BMC Optima Master Agent (generally referred to as the Peer Agent) provides an extensible SNMP agent capability that enables HP OpenView to manage ECS custom applications. Peer agent code is included in the custom ECS agent code and is part of the ECS standard custom code delivery.

- HP OpenView

  HP OpenView is a network management application extended by ECS to provide application management capabilities. The application provides a GUI for status monitoring of all ECS network devices and custom applications.

- Tivoli Software Services

  Tivoli Software Services are the Tivoli Enterprise Console, Tivoli Event Server, Tivoli Sentry, Tivoli Logfile Adapter, and the Tivoli Managed Region Server (TMR) described in the Network and Enterprise Management Framework Processes Table. The Tivoli/ Enterprise Console, Tivoli Sentry, and Tivoli Courier are parts of the TMR.

- IQ/Access

  IQ/Access is a report generation tool to write and generate standardized management reports for the MSS database.

- Netscape Enterprise Server

  The Netscape Enterprise Server implements a web interface to the Remedy Action Request System (ARS) enabling ECS users to submit trouble tickets to ARS and review the status of existing trouble tickets.

- Perl

  The Perl language is used to attach and detach the ASTER standard header for e-mail sent to and received from the ASTER GDS.

4.9.1 MCI Software Description - Network and Enterprise Management Component

The Management Software CSCI (MCI) is COTS and custom software enabling the Operations staff to monitor and coordinate the ECS services. The MCI is the following CSCs:

1. Network and Enterprise Management Framework
2. Security Service
3. Accountability Management
4. Trouble Ticket
5. Network Backup/Restore

4.9.1.1 Network and Enterprise Management Framework Functional Overview

The network and enterprise management framework monitors and controls the network, applications, and hosts distributed throughout the network. The framework is made of the HP OpenView Network Node Manager (NNM) and the Tivoli Enterprise Console (T/EC) COTS products. The OpenView NNM and the Tivoli Enterprise Console monitor and control the network and are the integration platform for other management tools, both custom and COTS, for a range of system management needs (e.g., network management, GUI development, and database usage).

Network Management Framework - OpenView

OpenView Network Node Manager (NNM) uses the Simple Network Management Protocol (SNMP) to monitor and control network objects. OpenView NNM has a "discovery" service that automatically detects network devices such as routers, bridges, and hosts and adds these devices to its database. Identification of other ECS managed objects is provided through ECS developed software in the Management Agent CSCI (MACI). The MACI CSCI informs the OpenView NNM of application elements. This information, along with mode management information, is collected by the OpenView NNM, saved in the map, object, and topology data views (tables) within the OpenView Management Database (DB) (in the Map DB, Object DB, and Topology DB views (tables)). This information is also used to build operations maps to show the logical layout and status of ECS managed objects. OpenView NNM also provides the capability to develop scripts that define action routines for each event received from each managed object (a device). OpenView NNM provides API support for integrating other management application packages and for developing custom management applications.

Enterprise Management Framework - Tivoli Enterprise Console

Tivoli monitors the performance of all managed hosts. The Tivoli Event server receives performance information from the Tivoli Sentry agent residing in each managed host, including free disk space, amount of swap space available, CPU usage, and number of active processes. When any of these metrics exceed a configurable threshold set by the Operations staff, Sentry sends the warning to the management station notifying the Operations staff of the potential system impact. Tivoli also supports the process scheduling on hosts in its management region to satisfy routine administrative tasks.

Fault Management Service

OpenView NNM and Tivoli provide basic fault management tools. The OpenView NNM receives general fault notifications from managed network devices via SNMP traps. Tivoli receives host performance threshold notifications when thresholds have exceeded an unacceptable level. The framework tools possess rules-based capability for reacting to faults, updating managed object operational status, and forwarding fault notices to the SMC.

Performance Management Service

Performance management is a task performed by the Operations staff. The OpenView NNM and Tivoli framework enable the Operations staff to collect and display resource usage trend

information. The NNM is used to monitor network device performance parameters such as packet throughput. The Tivoli monitors host resource usage thresholds and warns operations when they are exceeded.

Mode Management Service

The Mode Management Service (MMS) is an ECS developed service that is tightly integrated with HP OpenView. The MMS enables ECS applications to be configured into an operational mode and also provides support for ECS applications to be configured into training and testing modes during operations. The MMS incorporates the mode management user interface directly into the HP OpenView GUI and provides methods to activate and deactivate a mode. In addition, the MMS provides a mode specific user interface for accessing CSS lifecycle control (start-up and shutdown) commands. Monitoring capabilities are provided within HP OpenView and are enhanced to reflect mode specific status of software system, subsystem, application, program, and process level entities. Hardware is mode independent so its status is reflected within every mode in which it is configured. HP OpenView graphically supports multiple modes through the use of separate sub-maps and symbol labels. The map can have any number of sub-maps defined that decompose the basic high level map representation. Each mode has its mode specific map (and associated sub-maps) predefined to recognize and support the hardware and software items that are supporting the given mode.

4.9.1.2 Network and Enterprise Management Framework Context

Figure 4.9.1.2-1 is the Network and Enterprise Management Framework context diagram. The diagram shows the events sent to the Network and Enterprise Management Framework CSC and the events the Network and Enterprise Framework CSC sends to other CSCIs or CSCs. Table 4.9.1.2-1 provides descriptions of the interface events shown in the Network and Enterprise Management Framework context diagram.

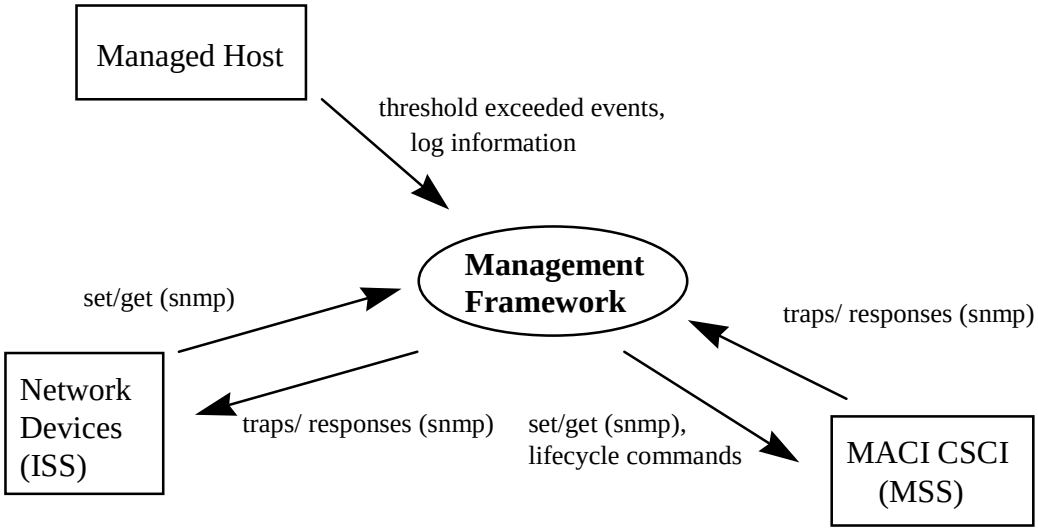

Figure 4.9.1.2-1. Network and Enterprise Management Framework Context Diagram

Table 4.9.1.2-1. Management Framework Interface Events

Event	Interface Event Description
set/get (snmp)	The SNMP set/get commands used by HP OpenView with network devices to set MIB variables and to get the value of a MIB variable, respectively. The MACI CSCI in each managed host also receives secure emulations via DCE Remote Procedure Calls.
traps/responses (snmp)	The SNMP traps/responses used by network snmp compliant devices to send event notifications and requested MIB variable responses to the OpenView NNM. The MSS management agent in each managed object sends secure emulations via DCE Remote Procedure Calls for responses sent by managed applications.
lifecycle commands	OpenView NNM issues startup/shutdown lifecycle commands via the MACI CSCI to applications in the managed hosts.
Threshold exceeded events	Managed hosts with the Tivoli Sentry agent residing in them provide host resource usage metrics to the Tivoli Enterprise Console.
Log information	Managed hosts with the Tivoli Sentry agent provide access to host log information.

4.9.1.3 Network and Enterprise Management Framework Process Architecture

Figure 4.9.1.3-1 is the Network and Enterprise Management Framework architecture diagram. The diagram shows the events sent to the Network and Enterprise Framework CSC processes and the events the Network and Enterprise Framework processes send to other processes.

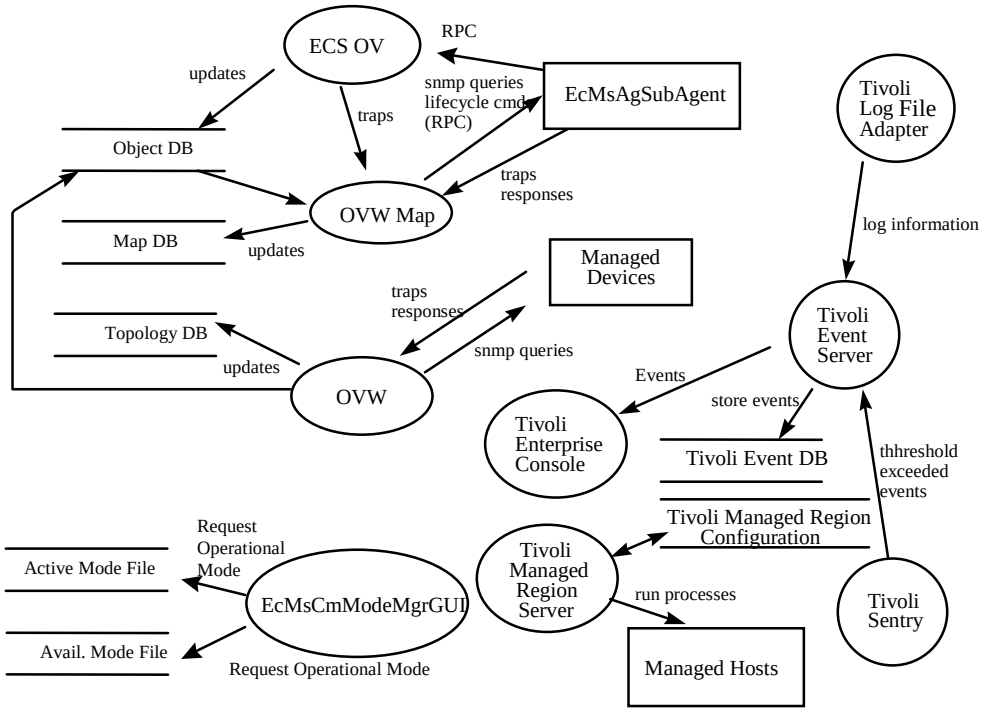

Figure 4.9.1.3-1. Network and Enterprise Management Framework Architecture Diagram

4.9.1.4 Network and Enterprise Management Framework Process Descriptions

Table 4.9.1.4-1 describes the processes in the Network and Enterprise Management Framework architecture diagram.

Table 4.9.1.4-1. Network and Enterprise Management Framework Processes (1 of 2)

Process	Type	COTS/ Developed	Functionality
Open View (OVW)	Daemon	COTS	The OVW process sends SNMP queries to managed objects in the network and uses the response information to update the OV Object DB and Topology DB and the OVW GUI. This process can also manipulate the MIB variable settings using the SNMP Set command.
OVW Map	GUI	COTS	The Open View for Windows (OVW) Map provides a graphical and hierarchical presentation of the managed objects including network devices, hosts, and ECS applications.
ECS OV	Daemon	Developed	The ECS OV process receives application status information sent by the EcMsAgSubAent to update the OV Object DB and Map DB.
Tivoli Event Server	Server	COTS	The Tivoli Event Server receives the events sent by Tivoli managed hosts and stores them in a proprietary database. (The database is proprietary, but is based largely on an earlier version of Sybase)
Tivoli Sentry	Server	COTS	The Tivoli Sentry resides on each managed host and monitors system resource usage. Tivoli Sentry sends threshold exceeded events to the Tivoli Event Server whenever usage goes above a configurable level.
Tivoli Enterprise Console (T/EC)	GUI	COTS	The T/EC provides the user with notifications if a threshold is exceeded. Examples include: CPU usage exceeding 97% - Critical Event Disk usage exceeds 95% - Critical Event Swap Space available is below 10 Megabytes (MB) – Warning Event The AutoSys daemon has become unavailable – Critical The string “REPEATED LOGIN FAILURES” was found in the syslog – WARNING Event
Tivoli Log File Adapter	Other	COTS	The Tivoli Logfile Adapter resides on managed hosts and monitors log files for predetermined strings. By default, the System Log (SYSLOG) file is monitored and configurations can be manipulated via adapters through tools provided by Tivoli.
Tivoli Managed Region (TMR) Server	Server	COTS	The Tivoli Managed Region Server resides at the management station. The Tivoli Managed Region Server communicates with client hosts in the Tivoli management region. The server maintains the status of these hosts and is capable of scheduling the running of scripts on these hosts for routine administration.

Table 4.9.1.4-1. Network and Enterprise Management Framework Processes (2 of 2)

Process	Type	COTS/ Developed	Functionality
EcMsCmModeMgrGUI	GUI	Developed	The Mode Management GUI provides a means for operations to define modes and initiate and control applications in all of the defined modes.
EcMsAgSubAgent	Other	Developed	The EcMsAgSubAgent sends the lifecycle commands (startup and shutdown) to the Hp OpenView applications and makes remote procedure calls to the ECS OV and traps and responses to the OVM Map process.

4.9.1.5 Network and Enterprise Management Framework Process Interface Descriptions

Table 4.9.1.5-1 provides descriptions of the Network and Enterprise Management Framework interface events shown in the Network and Enterprise Management Framework architecture diagram.

Table 4.9.1.5-1. Network and Enterprise Management Framework Process Interfaces (1 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Traps	One per trap state change	snmp / Trap protocols	*Process:* ECS OV	SNMP Traps are sent by managed network devices, hosts, and managed applications notifying of a change in the object state. The information is used to update the status of objects in OpenView databases.
Snmp queries	One per Snmp query	snmp “gets”	*Process:* OVW Map	HP OpenView sends SNMP queries to managed devices and to the EcMsAgSubAgent for information available in the MIB variables.
Updates	One per update	OV API calls	*Processes:* ECS OV, OVW Map, OVW	The OV Object DB, Topology DB, and Map DB are updated via the OV API.
Lifecycle cmds	One per lifecycle command	OV API calls, DCE RPC	*Process:* OVW Map	Operator start-up/shutdown commands to applications.
Responses	One per response	DCE RPC	*Process:* EcMsAgSubAgent	SNMP query answers from managed objects. To promote security, DCE Remote Procedure Calls (RPCs) are used instead of SNMP to communicate with managed applications. The RPC is between the OpenView management station and the subagents on each host.

Table 4.9.1.5-1. Network and Enterprise Management Framework Process Interfaces (2 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Request Operational Mode	One per request of operational mode	Rogue Wave file access API	*Process:* EcMsCmMode MgrGUI	The Mode management GUI updates an Active Modes file that is NFS mounted across all managed servers.
threshold exceeded events	One per threshold event exceed	Tivoli oserv daemon	*Process:* Tivoli Sentry	Events are sent to the Tivoli Event Server running on the Tivoli Management Region (TMR) Server.
store events	One per store events	Internal Tivoli API call	*Process:* Tivoli Event Server	Events received by the Tivoli Event Server are processed and stored in the Event database.
Events	One per events	Tivoli Event Console	*Process:* Tivoli Event Server	The Tivoli Enterprise Console retrieves events from the Event database and displays them to the M&O staff.
run processes	One per run processes	Internal Tivoli API call	Tivoli Managed Hosts	The TMR server is able to execute scripts on managed hosts. An example of this would be to periodically remove core files.
log informa-tion	One per log information	Internal Tivoli API call	*Process:* Tivoli Event Server	Events can be logged to a text file if configured within the event definition / configuration.

4.9.1.6 Network and Enterprise Management Framework Data Stores

Table 4.9.1.6-1 provides descriptions of the data stores used in the Network and Enterprise Management Framework architecture diagram.

Table 4.9.1.6-1. Network and Enterprise Management Framework Data Stores (1 of 2)

Data Store	Type	Functionality
Object DB	Database	The Object database contains all the objects (physical and logical) in the network that have been discovered by OpenView NNM.
Map DB	Database	The Map database stores presentation information for each object stored in the object database. A map is a collection of objects from the Object database along with their relationships. A map contains a subset of all the objects in the Object database.
Topology DB	Database	The Topology database contains an electronic representation of the topology of the infrastructure of the network. This includes all entities with IP addresses.
Tivoli Management Region Configuration	File	The Tivoli Management Region (TMR) Configuration defines the managed network host configuration and the TMR configuration is defined via an initialization procedure.

Table 4.9.1.6-1. Network and Enterprise Management Framework Data Stores (2 of 2)

Data Store	Type	Functionality
Event DB	Database	The Event DB contains those events forwarded to the tivoli event server from managed hosts. These events can be retrieved and displayed by the M&O staff for review on the T/EC.
Avail. Mode File	File	Available Mode File contains the modes defined for use in the ECS and that can be changed through the Mode Management GUI.
Active Mode File	File	Active Mode File contains those modes that are activated.

4.9.2 MCI - Security Functional Component

4.9.2.1 Security Functional Overview

Security Service monitoring in the ECS is accomplished through several commercial and public domain programs. The programs vary from aiding in administration of DCE, assisting the user in choosing a password difficult to break, monitoring key system files for signs of tampering and probing hosts for well known security violations.

4.9.2.2 Security Context

Figure 4.9.-2.2-1 is the Security Service context diagram. The diagram shows the events sent to the Security Service CSC from the operating system, communications devices, and the Operations staff and the events the Security Service CSC sends to the operating system, communications devices and the Operations staff. Table 4.9.2.2-1 provides descriptions of the interface events shown in the Security Service context diagram.

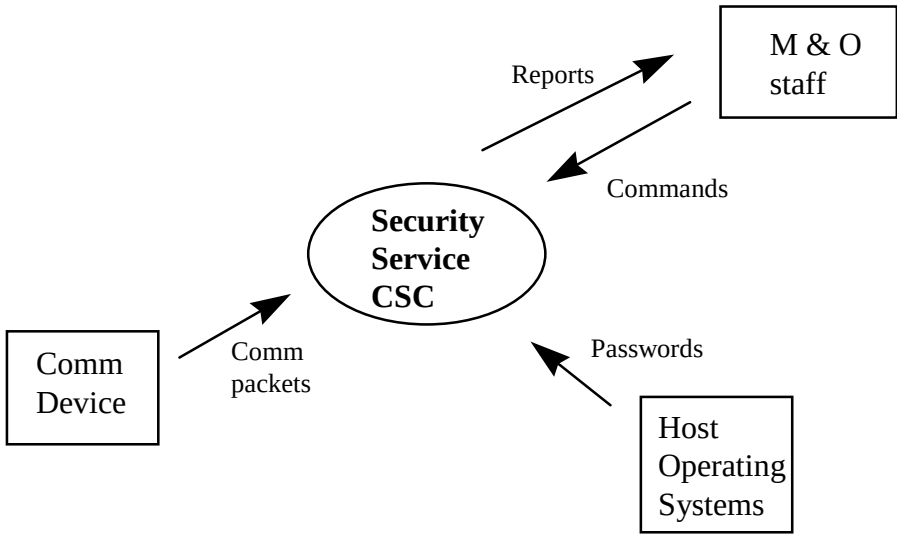

Figure 4.9.2.2-1. Security Service CSC Context Diagram

Table 4.9.2.2-1. Security Service CSC Interface Events

Event	Interface Event Description
Commands	The M&O staff issues commands to the Security Service CSC utilities to exercise system security setup.
Reports	The Security Service CSC utilities perform their functions and report results.
Comm packets	A packet reaches the ECS host from either an external source or from a host within the same site. The Security Service CSC analyzes packets for authorized sending sources.
Passwords	A password list is obtained from the Network Information Service (NIS) master by issuing a ypcat passwd command. This list is analyzed to see if decryption of a password is possible.

4.9.2.3 Security Architecture

Figure 4.9.2.3-1 is the Security Service CSC architecture diagram. The diagram shows the events sent to the Security Service CSC processes and the events the Security Service CSC processes send to other processes.

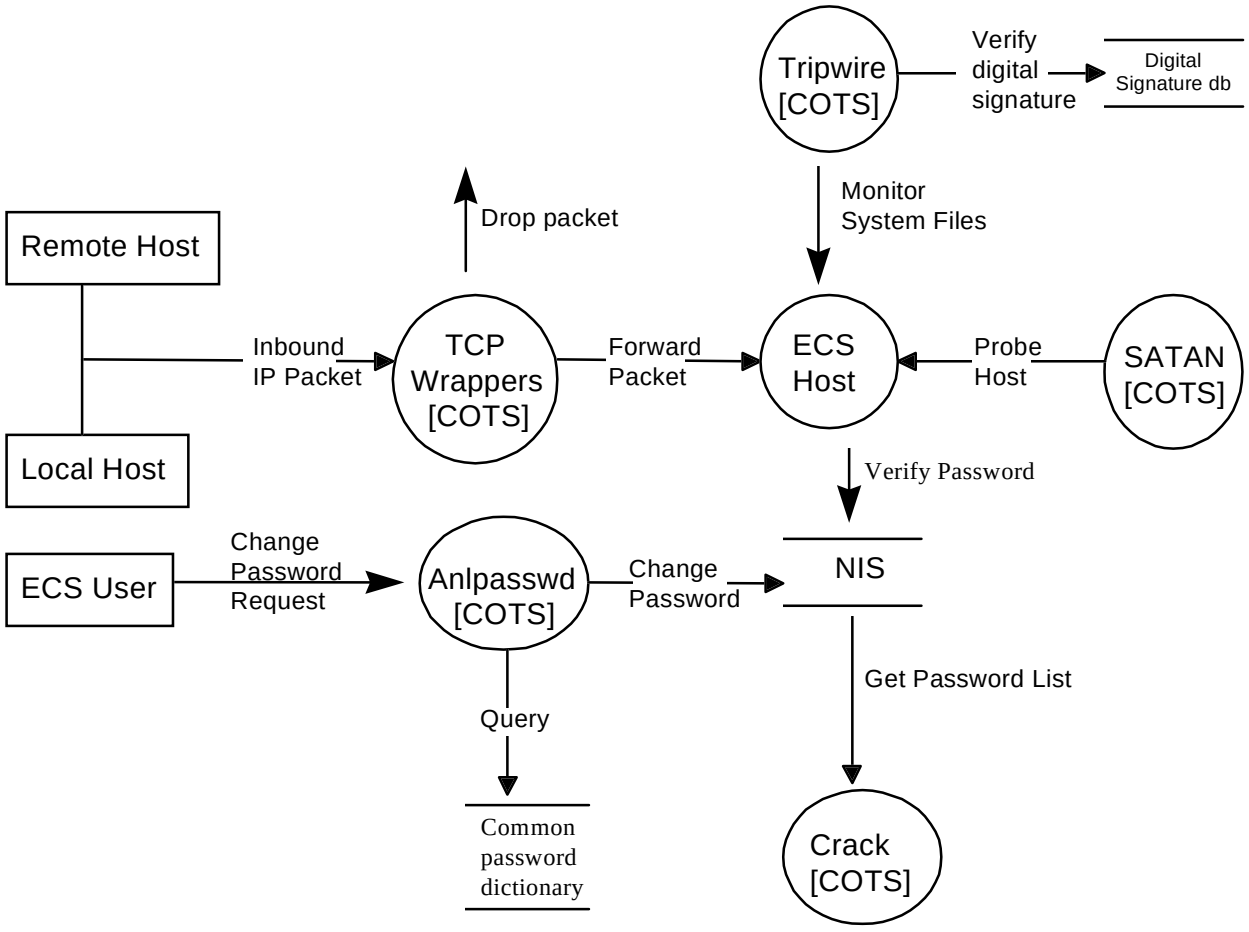

Figure 4.9.2.3-1. Security Service CSC Architecture Diagram

4.9.2.4 Security Process Descriptions

Table 4.9.2.4-1 provides descriptions of the processes shown in the Security Service CSC architecture diagram.

Table 4.9.2.4-1. Security Service CSC Processes

Process	Type	COTS / Developed	Functionality
anlpasswd	Other	COTS	Anlpasswd is a replacement for the standard UNIX passwd and yppasswd programs. Anlpasswd provides functionality by checking the selected user password to determine if the password is common or trivial and easy to break.
TCP Wrappers	Other	COTS	TCP Wrappers verifies the origin of incoming IP packets from an authorized host for services TCP Wrappers can filter. TCP Wrappers runs on each ECS host (UNIX) at a specific site.
Tripwire	Other	COTS	Tripwire periodically verifies that system files have not been altered. Tripwire is able to catch modifications by verifying the current and stored digital signatures of the command.
SATAN	GUI / Other	COTS	SATAN is run by an M&O staff member periodically to determine if any common vulnerabilities exist on ECS controlled hosts. The results are displayed in a web browser upon completion of a scan.
Crack	Other	COTS	An M&O staff member runs Crack periodically to search for passwords that can be broken and were not caught by anlpasswd. These passwords are added to the anlpasswd dictionary or a new rule can be added to eliminate this error.

4.9.2.5 Security Process Interface Descriptions

Table 4.9.2.5-1 provides descriptions of the interface events shown in the Security Service CSC architecture diagram.

Table 4.9.2.5-1. Security Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Inbound IP packet	One per inbound IP packet	TCP/IP protocols	TCP/IP protocols	A packet reaches the ECS host from either an external source or from a host within the same site.
Forward Packet	One per forward packet	Inetd – UNIX daemon	TCP Wrappers	If the IP header indicates the packet originates from a host that has not been blocked by TCP Wrappers, the packet is forwarded via the appropriate internet service to an ECS Host.

Table 4.9.2.5-1. Security Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Drop packet	One per drop packet	N/A	TCP Wrappers	If the IP packet originates from a disabled source, it is dropped and no further processing is done on the packet.
Monitor system files	One per monitor system files	crontab	Tripwire	Critical system files are watched periodically for changes in their digital signature that could signal a maliciously altered system file or service.
Verify digital signature	Once per verify digital signature	Tripwire internal call	Tripwire	The newly computed digital signature of a file is verified against the stored historical copy of the same file.
Probe host	One per probe host	Netscape interface	SATAN	SATAN is used on an ad hoc basis to probe the hosts within a network to determine if any common security violations exist.
Get password list	One per get password list	NIS system call	Crack	A password list is obtained from the NIS master by issuing a ypcat passwd command. This list is run through crack to see if crack is able to decrypt any user's password.
Change password request	One per change password request	Command line	Command line	An ECS user attempts to change their password and the request is verified by anlpasswd that the new password does not contain any trivial or easy to guess password.
Query	One per query	NIS system call	Anlpasswd	Check the common word dictionary to ensure the attempted new password is not in this list.
Change password	One per change password	NIS system call	Anlpasswd	After the new password passes the anlpasswd validation process, a request is sent to the NIS master to modify the user's password.
Verify password	Once per verify password	NIS system call	Anlpasswd	A request is sent from the ECS host to NIS to verify that a login request is valid.

4.9.2.6 Security Data Stores

Table 4.9.2.6-1 provides descriptions of the Security Service CSC data stores shown in the Security Service CSC architecture diagram.

Table 4.9.2.6-1. Security Data Stores

Data Store	Type	Functionality
NIS database	Other	This UNIX service enables a common login on a number of machines and mapping for a user's Network File System (NFS) mounted home directory. The passwd map stores a user's login id, group id, and password in the NIS database.
Common word dictionary	Other	This sorted text file contains common words used by a user as a password. Anlpasswd verifies that the new password change does not include a word listed in the Anlpasswd file.
File signature database	Other	This proprietary database is used by Tripwire to record the digital signature for each system file it monitors.

4.9.3 MCI - Accountability Management Component

4.9.3.1 Accountability Management Functional Overview

The Accountability Management Service supports User Registration and Order Tracking.

User Registration

ECS provides for two generic classes of users: guest users and registered users. Guest users are not formally registered. Registered users have submitted requests for a registered user account and have accounts, based on an approval process. Registered users can access services and products beyond those available to guest users.

Guest users can submit a request for a registered user account. The submitted request is captured in a database of pending requests. The Operations staff accesses the database of pending requests and creates registered user accounts for approved requests.

The user registration server supports the creation, modification and maintenance of profiles for each registered user. The user profile is replicated at each DAAC. Each DAAC is capable of browsing foreign user profiles, but only capable of modifying user profiles created within the DAAC.

The user registration GUI enables the DAAC Operations staff to view user requests and user profiles for modification. The user profile information includes the user's name, identification code, primary DAAC, organizational affiliation, investigating group (such as an instrument team) affiliation (if any), assigned project, mailing address, shipping address for data or product order distribution media preferences for product orders, telephone number and electronic mail address (if any).

The Accountability Management Service enables the various subsystems to request user profile information such as the user's electronic mail address and the shipping address for product order or data distribution.

Order Tracking

The Order Tracking service provides the capability to track a product order's status during request processing. The Order Tracking service centralizes order status in the MSS database instead of going to each subsystem to collect it. The Order Tracking Server provides the public interface to other subsystems for updating order and request status in real time. The Order Tracking GUI enables order status browsing by user name, orderId, or the orderId's associated requestId.

All RPCs to the Accountability servers are authenticated via the Access Control List (ACL) Database.

Order Tracking has interfaces with other subsystems to provide access to order and request status. This tracking information is saved in the Order Tracking database.

4.9.3.2 Accountability Management Context

Figure 4.9.3.2-1 is the Accountability Management Service context diagram. The diagram shows the events sent to the Accountability Management Service CSC and the events the Accountability Management Service CSC sends to other CSCIs or CSCs. Table 4.9.3.2-1 provides descriptions of the interface events shown in the Accountability Management Service CSC context diagram.

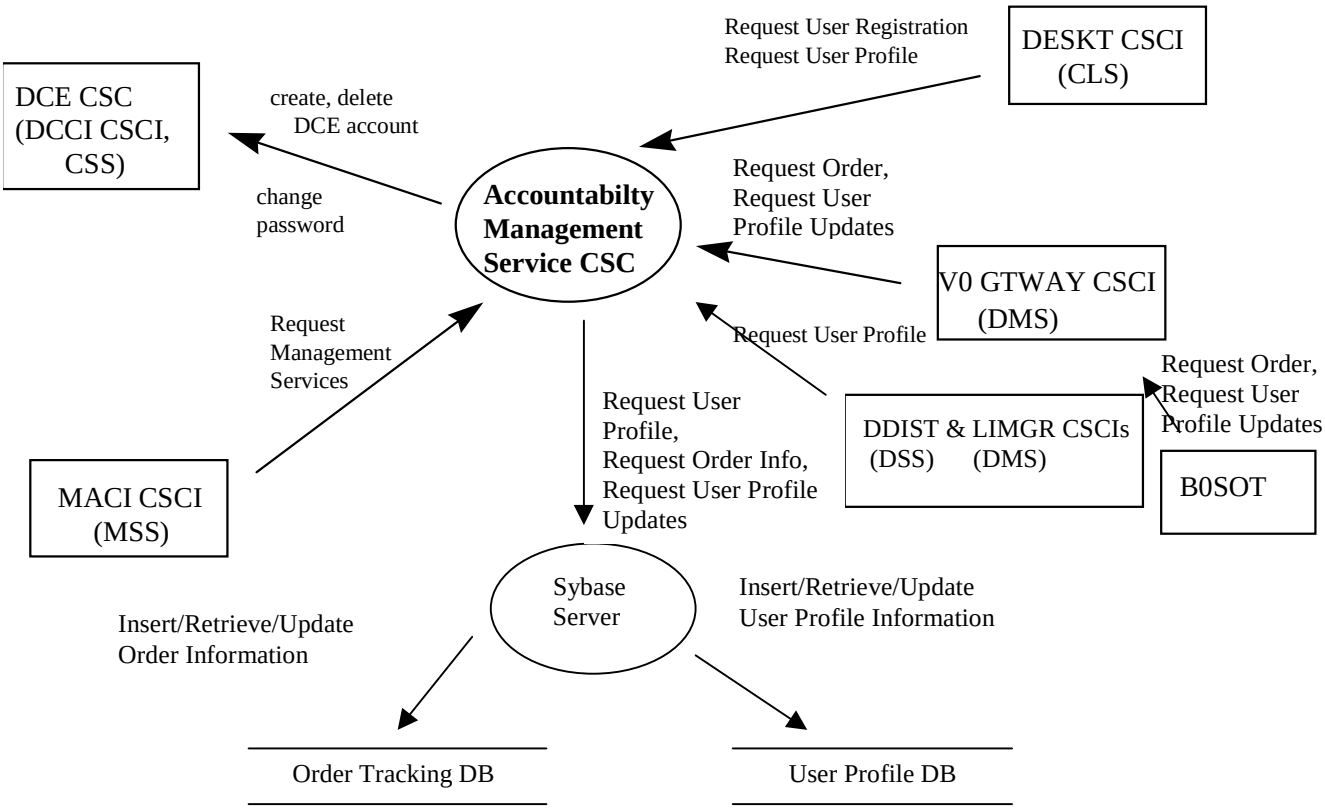

Figure 4.9.3.2-1. Accountability Management Service Context Diagram

Table 4.9.3.2-1. Accountability Management Service Interface Events (1 of 2)

Event	Interface Event Description
Request user profile updates	Users can update their User Profile through the B.0 Search and Order Tool (B0SOT). This includes their addresses (user, shipping, billing, and e-mail) and other pertinent profile information.
Request user profile	The User Registration Server has an interface to provide user profile information to other CSCIs or CSCs from other subsystems or B0SOT. The user profile is retrieved by either a user Id or an ECS Authenticator.
Request user registration	Guest users submit ECS registration requests through the DESKT CSCI or B0SOT.
Create, delete DCE account, change password	The User Registration Server has an interface to the DCE security server to create a DCE user, delete a DCE user and to change a user's DCE password by request.

Table 4.9.3.2-1. Accountability Management Service Interface Events (2 of 2)

Event	Interface Event Description
Insert/Retrieve/Update User Profile Information	The Sybase Server inserts, retrieves, or updates (create/modify/delete) system default profile information in the database by request.
Request Order (Info)	The Order Tracking Server receives order requests from the B0SOT via the V0 GTWAY CSCI. See specifically the DMS role in the Request Management Services event interface description.
Request management services	The MACI CSCI provides a basic management library of services to the CSCIs/CSCs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run. The MACI CSCI also interfaces with other CSCIs/CSCs to perform the following: • **DMS Order/Request tracking update** - The V0 GTWAY CSCI interfaces with Accountability Management Service Order/Request Tracking service to create a user product order. • **User Profile Request** - The Accountability Management Service provides requesting CSCIs/CSCs with access to User Profile parameters such as e-mail address and shipping address to support their processing activities.
Insert/ Retrieve/Update Order Information	The Sybase Server retrieves or updates (create/modify/delete) product order status information in the database by request.
Retrieve Configuration Parameters	The Sybase Server retrieves or updates system default parameters (i.e., file sizes, file thresholds, number of retries) contained within subsystem configuration files.

4.9.3.3 Accountability Management Architecture

Figure 4.9.3.3-1 is the Accountability Management Service architecture diagram. The diagram shows the events sent to the Accountability Management Service CSC processes and the events the Accountability Management Service CSC processes send to other processes.

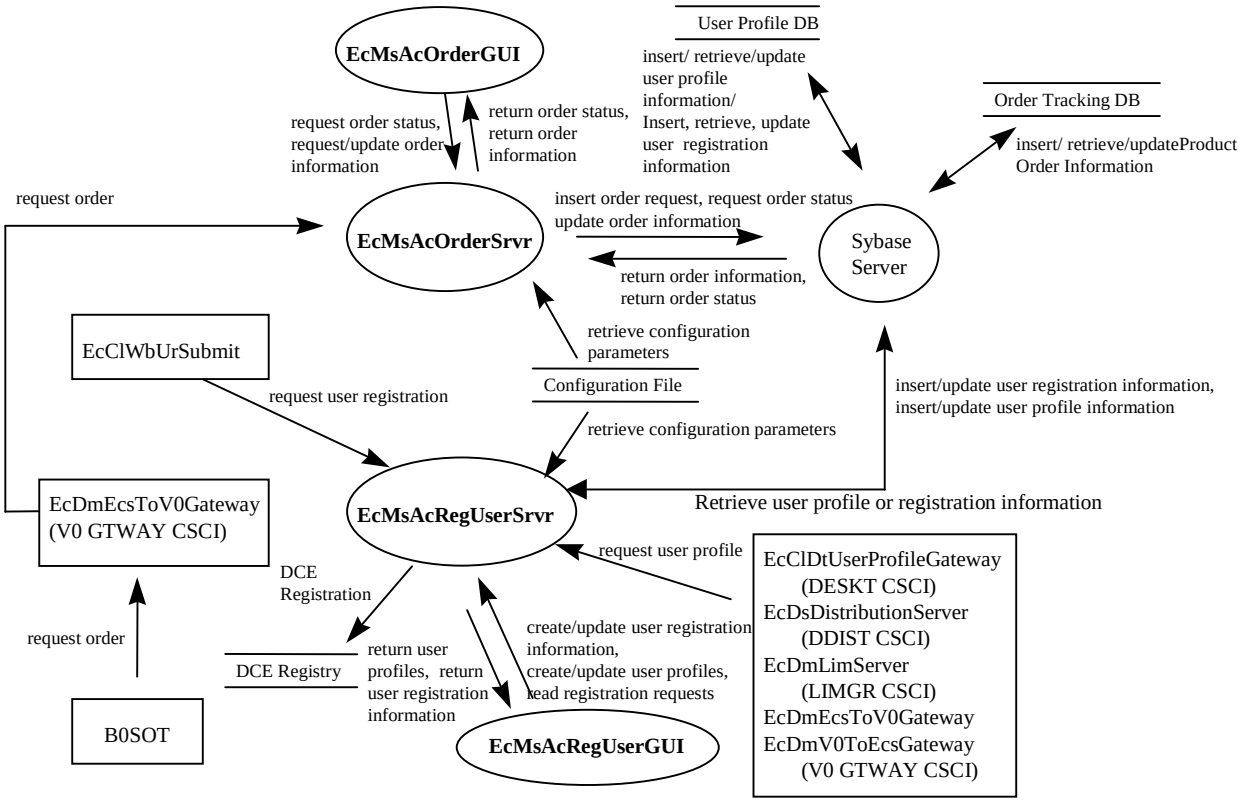

Figure 4.9.3.3-1. Accountability Management Service Architecture Diagram

4.9.3.4 Accountability Management Processes

Table 4.9.3.4-1 provides the descriptions of the processes shown in the Accountability Management Service architecture diagram.

Table 4.9.3.4-1. Accountability Management Processes (1 of 2)

Process	Type	COTS / Developed	Functionality
EcMsAcRegUserSrvr	Server	Developed	The User Registration Server provides an internal interface to the User Registration GUI and an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve user request 2. Insert, delete, update, retrieve user profile 3. Insert, delete, update, retrieve registered user 4. Retrieve a list of user requests 5. Retrieve a list of user profiles 6. Retrieve a list of registered users 7. Change DCE password 8. Change V0 gateway password The EcMsAcRegUserSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Request processing de-coupled from an RPC thread • Multiple threads within a single request
EcMsAcRegUserGUI	GUI	Developed	The User Registration graphical user interface enables the viewing and updating of user profiles. The GUI enables the user to : 1. Add a ECS user and send e-mail notification 2. Delete a ECS user 3. Modify a ECS user profile 4. Change a DCE password 5. Change the V0 gateway password 6. Change ASTER category and send e-mail 7. Change the DAR privilege The ASTER e-mail address described above is stored in the Accountability configuration file. The Accountability configuration file is read in when the Accountability GUI is started up.

Table 4.9.3.4-1. Accountability Management Processes (2 of 2)

Process	Type	COTS / Developed	Functionality
EcMsAcOrderSrvr	Server	Developed	The Order Tracking Server provides an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve order 2. Insert, delete, update, retrieve request 3. Retrieve a list of orders 4. Retrieve a list of requests 5. Update order status 6. Update request status The EcMsAcOrderSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Request processing de-coupled from an RPC thread • Multiple threads within a single request
EcMsAcOrderGUI	GUI	Developed	This graphical user interface enables the user to retrieve the order and request from the Accountability database. The following functions are available: 1. Retrieve order and request by order id or request id. 2. Retrieve order and request by user name. 3. Retrieval can be filtered by order and request status.
Sybase Server	Server	COTS	The SQL server supporting access to the Sybase DBMS. The interface between processes and the databases for storage and retrieval of data or information.

4.9.3.5 Accountability Management Process Interface Descriptions

Table 4.9.3.5-1 provides descriptions of the interface events shown in the Accountability Management Service architecture diagram.

Table 4.9.3.5-1. Accountability Management Service Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Request user registration	One per request for user registration	*Libraries:* MsAcClnt MsAcComm	*Process:* EcClWbUr	A guest user requests to become a registered ECS user.
DCE registration	One per DCE registration	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserGUI	User registration for DCE interaction.
Request user profile	One per user profile request	*Libraries:* MsAcClnt MsAcComm	*Processes:* EcClDtUserProfileGateway, EcDsDistributionServer, EcDmLimServer, EcDmEcsToV0Gateway, EcDmV0ToEcsGateway	A request for user profile information for viewing or modification.
Insert/update user registration information	One per user registration insert/ update.	*Libraries:* MsAcClnt MsAcComm	*Processes:* EcMsAcRegUserGUI, EcClDtUserProfileGateway	User registration information is added to or modified in the User Profile database (DB).
Insert/update user profiles	One per user insert/ update profile	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserSrvr	User profiles are added to or modified in the User Profile database (DB).
read registration requests	One per read registration request	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserGUI	A user request, via the EcMsACRegUserGUI, to retrieve a registration request for action or to modify the registration request.
create/update user profiles	One per create/ update user profile	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserGUI	Create or modify user profile information by user request via the EcMsAcRegUserGUI.
create/update user registration information	One per create/update user registration	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserGUI	Create or modify user registration information by user request via the EcMsAcRegUserGUI.

Table 4.9.3.5-1. Accountability Management Service Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
insert/retrieve/update user profiles	One per insert/retrieve/update of user profile	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	User Profile information is retrieved for viewing or updated by request via the Sybase Server.
insert/retrieve/update user registration information	One per insert/retrieve/update of user registration information	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	User registration information is stored or retrieved by request via the Sybase Server.
return user registration information	One per return of user registration request	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	User registration information is returned to the requester at the GUI.
return user profiles	One per return of user provile	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	User profile information is returned to the requester at the GUI.
retrieve configura-tion parameters	One per retrieve of configuration parameters	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcRegUserSrvr	The default parameters are retrieved upon startup by the servers or can be changed by request and the servers restarted.
Request order	One per Request order	*Libraries:* MsAcClnt MsAcComm	B0SOT Tool Process: EcMsAcOrderSrvr	A request by the user via B0SOT for a product order via the EcDmV0ToECSGateway to the EcMsAcOrderSrvr.
request order status	One per request order status	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderSrvr	The EcMsAcOrderGUI obtains current order status from the Order Tracking Database via the Sybase Server.
request order	One per request order	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderSrvr	A request by the user via B0SOT for a product order.
request/update order Information	One per request/update order information	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderGUI	Order information is retrieved for viewing or updated by the Operations staff via the Sybase Server.

Table 4.9.3.5-1. Accountability Management Service Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
retrieve configura-tion parameters	One per retrieval of configuration parameters	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderSrvr	The configuration parameters are retrieved upon startup by the servers or can be changed by request and the servers restarted.
Insert order request	One per insert order request	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderSrvr	The product order request is inserted into the Order tracking database (DB).
Update order information	One per update of order information	*Libraries:* MsAcClnt MsAcComm	*Process:* EcMsAcOrderGUI	The operations staff requests an update of product order information.
return order information	One per return of order information	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	Product order information is returned per operations request.
return order status	One per return order status	*Libraries:* MsAcClnt MsAcComm	*Process:* Sybase Server	Product order status is returned per operations request.

4.9.3.6 Accountability Management Data Stores

Table 4.9.3.6-1 provides descriptions of the data stores shown in the Accountability Management Service architecture diagram.

Table 4.9.3.6-1. Accountability Management Data Stores

Data Store	Type	Description
User Profile DB	Database	The User Profile DB contains requests for user registrations and it also contains the profile information including mailing addresses, e-mail address, and project affiliations of approved registered users.
Order Tracking DB	Database	The Order Tracking DB contains product orders and user requests with the associated current processing status.
DCE Registry	Other	This file contains the user DCE registration parameters.
Accountability Configuration File	Other	The Accountability software obtains configuration parameters from a configuration file at startup. It contains: 1. Host Name 2. Database login information 3. Application log level 4. Application log size 5. Hp OpenView start up scripts 6. ASTER e-mail address

4.9.4 MCI - Trouble Ticket Component

4.9.4.1 Trouble Ticket Functional Overview

Remedy's Action Request System (ARS), commonly referred to as Remedy, implements the Trouble Ticketing service in the ECS. The GUI provided with Remedy enables the Operations staff to enter and track trouble tickets affecting both local and ECS system-wide resources. In addition, a custom web-based interface using the Remedy API enables ECS registered users to submit new trouble tickets and to obtain the current resolution status of their open trouble tickets. The delivered configuration of Remedy includes trouble escalation policies, operator notifications, and status reports to aid in the problem resolution process.

4.9.4.2 Trouble Ticket Context

Figure 4.9.4.2-1 is the Trouble Ticket context diagram. The ARS receives new trouble tickets from users. In addition, new trouble tickets are created from existing trouble tickets forwarded by other DAACs or an external system such as ASTER GDS, NSI, or Landsat 7. The ARS stores information in several Sybase tables – one table per schema used by the ARS. Notifications are automatically sent to the appropriate administrators upon creation and closure. An alarm notification is also sent if a trouble ticket has not been assigned to an investigator within a predetermined time period determined by ECS policy and procedures. The user who submits a Trouble Ticket is automatically notified upon creation of a trouble ticket and upon closure of the trouble ticket. Additionally, a Remedy mail daemon is available to receive trouble tickets via e-mail that are forwarded from other DAACs.

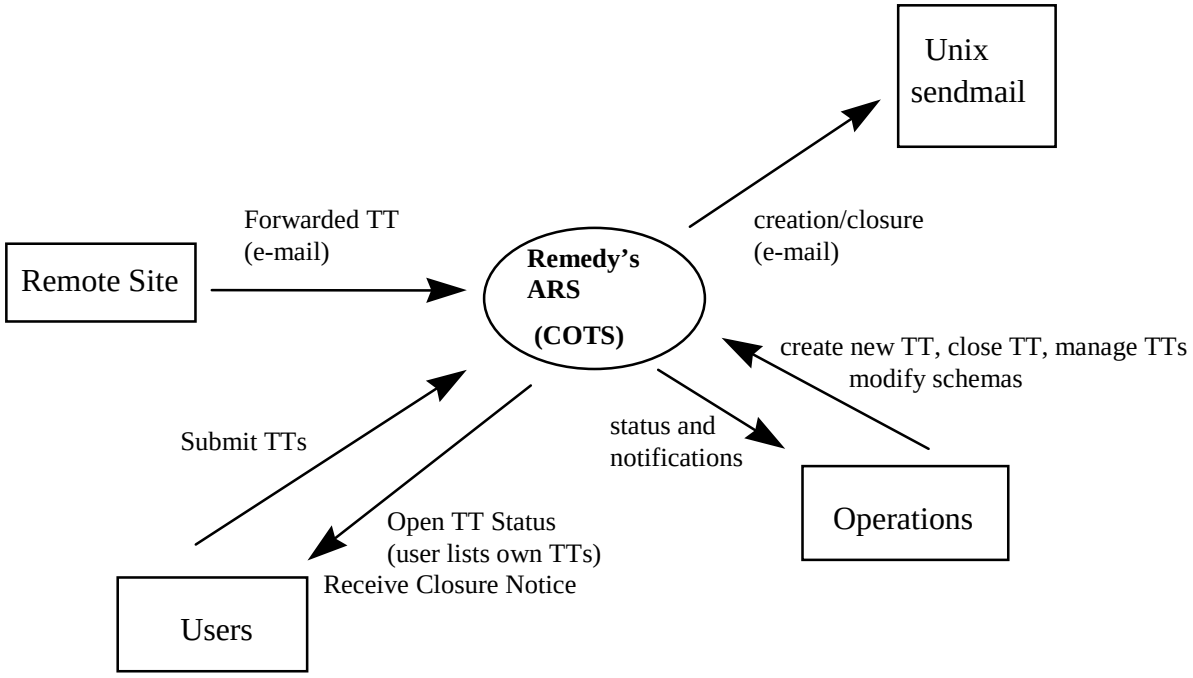

Figure 4.9.4.2-1. Trouble Ticket Context Diagram

Table 4.9.4.2-1 provides descriptions of the interface events shown in the Trouble Ticket context diagram.

Table 4.9.4.2-1. Trouble Ticket Interface Events

Event	Interface Event Description
Status and Notifications	The M&O Staff and the Trouble Ticket (TT) Administrator can query all information concerning a particular TT through Remedy's aruser GUI. Notifications are sent to the staff member or group responsible for a particular stage of a Trouble Ticket. These notifications can include warnings that a TT has been assigned to a staff member or that a TT has been left in a particular state for too long. Notifications can be sent either by e-mail or the Remedy notifier GUI.
Create New TT	The M&O staff is able to submit new TTs using the aruser GUI supplied with the COTS software package.
Close TT	Upon resolution of a Trouble Ticket, the M&O staff member annotates the corrective actions in the TT schema and moves the TT to a Closed state. This triggers a Receive Closure Notice action.
Receive Closure Notice	An e-mail message is sent to the originator after the TT has been closed. This message includes the TT ID number and corrective actions taken.
Submit TT	An ECS user can submit a TT via the custom web interface. This interface enables interaction with the User Profile Server. This is an alternative to calling the DAAC directly.
Open TT status (user lists own TTs)	An ECS user can query the TT database and find the current status of opened tickets.
Manage TTs	The TT administrator assigns open TTs to the appropriate M&O staff member. The M&O staff member receives notification by e-mail or the notifier tool based on preferences set in the Remedy User schema for that administrator.
Modify Schemas	The TT administrator modifies schemas and screen layouts. This is not encouraged as it can produce incompatible TTs with other site schemas. Also, trouble ticket priority escalations and filters used by the TT administrator to determine escalations can be altered by the TT administrator.
TT Forwarded from other site	Using the mail template of the Remedy mail daemon (armaild), sites can create and forward new TTs to another site. This new TT has the original ID stored as a Unique Identifier.

4.9.4.3 Trouble Ticket Architecture

Figure 4.9.4.3-1 is the Trouble Ticket architecture diagram. The diagram shows the events sent to the Remedy Action Request System (ARS) COTS process and the events the Remedy ARS COTS process sends to other processes (Remedy GUIs, daemons, and the Sybase Server).

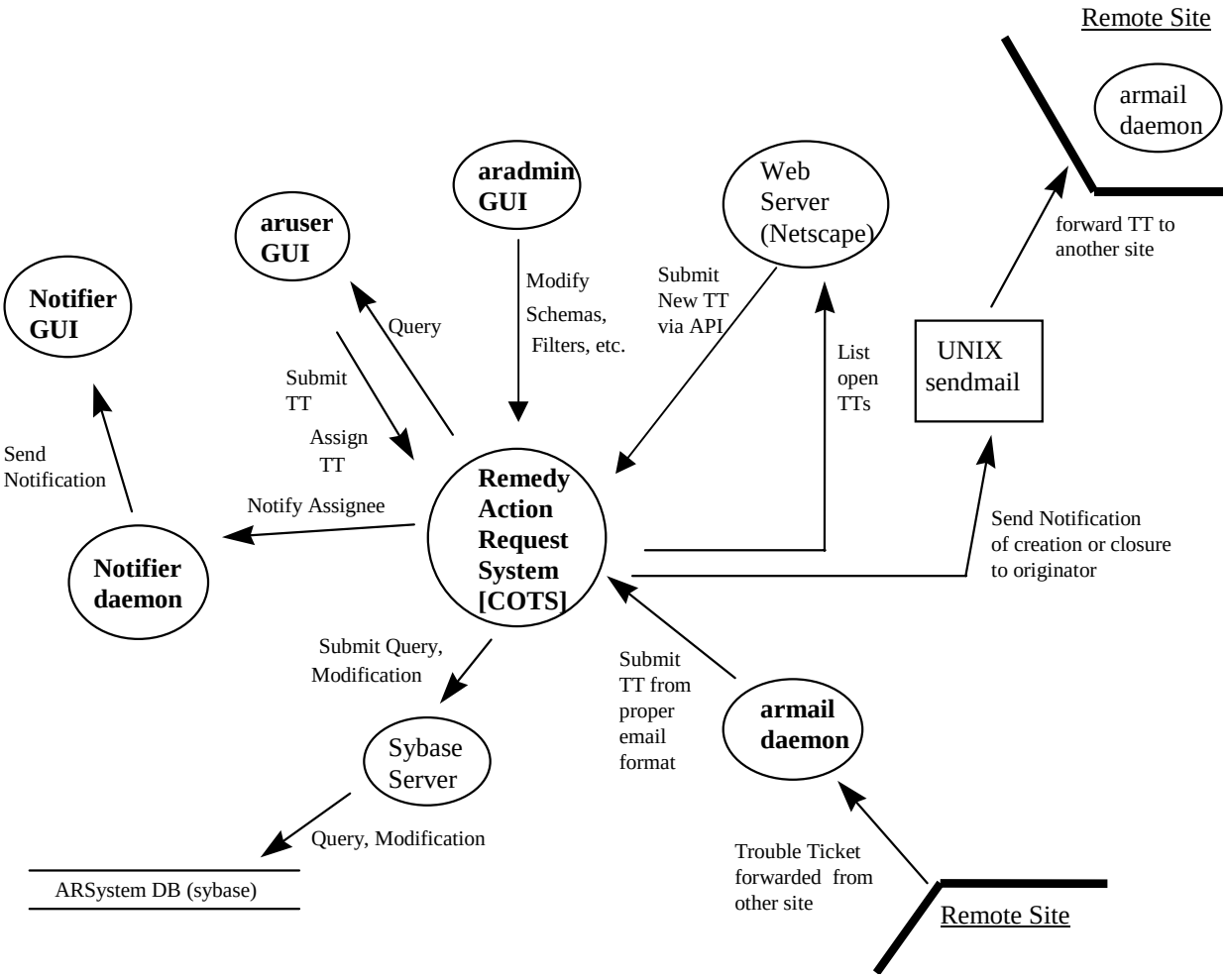

Figure 4.9.4.3-1. Trouble Ticket Architecture Diagram

4.9.4.4 Trouble Ticket Process Descriptions

Table 4.9.4.4-1 provides descriptions of the processes shown in the Trouble Ticket architecture diagram.

Table 4.9.4.4-1. Trouble Ticket Processes

Process	Type	COTS / Developed	Functionality
aruser GUI	GUI	COTS	The aruser GUI enables the M&O staff member to: 1. Submit a new TT 2. Query information about an existing TT 3. Move a TT to a Closed state and annotate the resolution The TT admin uses this GUI to: 1. Add / Modify administrators in Remedy's User schema 2. Assign TTs to M&O staff
aradmin GUI	GUI	COTS	The TT administrator uses this GUI to: 1. Update schemas and aruser screen layouts 2. Update escalation policies and filters
Web Server	Server	Developed	This interface enables the ECS user access to the Trouble Ticket process without directly contacting an M&O staff member. The user is able to: 1. Submit a new TT 2. Query the status of an existing TT they submitted
notifier GUI	GUI	COTS	This tool provides notification upon submission of a new TT or when a M&O staff member is assigned responsibility for a TT.
notifier daemon	Server	COTS	The notifier daemon sends notifications to an M&O staff member's notifier GUI or by e-mail if the staff member's Remedy notification preference is set to e-mail.
Remedy ARS	Server	COTS	The Remedy ARS interacts with its associated GUIs via the provided Remedy daemons and the ARSystem DB. Error messages are logged to an aerror.log log file.
armail daemon	Server	COTS	The armail daemon monitors a mailbox (/var/spool/mail/arsystem) for incoming TTs formatted in the proper Remedy layout for TTs. Upon reception of a valid, formatted message, a new TT is created.
UNIX sendmail	Server	COTS	The sendmail daemon is an integral part of Remedy and must be properly configured for both e-mail notifications and TT forwarding to be accomplished.

4.9.4.5 Trouble Ticket Process Interface Descriptions

Table 4.9.4.5-1 provides descriptions of the interface events shown in the Trouble Ticket architecture diagram.

Table 4.9.4.5-1. Trouble Ticket Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Submit TT	One per submit TT	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* aruser GUI	A new Trouble Ticket is created by M&O staff and entered into the Remedy system. A notification is sent to the TT administrator of the existence of a new TT.
Assign TT	One per assign TT	*Program:* MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	After receiving the notification, the TT administrator assigns the TT to an M&O staff member.
Notify Assignee	One per notify assignee	*Program:* MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	A request is sent to the Remedy notifier daemon to notify the M&O staff member of responsibility for the TT.
Send Notification	One per send notification	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* Notifier daemon	The notifier daemon notifies the responsible M&O staff member via the notification GUI or by e-mail depending on User schema settings.
Query	One per query	*Program:* MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	The M&O staff member, upon receiving notification of a new TT, queries the Remedy TT schema to find the detailed information and process the TT.
Submit, Query, Modification,	One per submit, query, or modification	*Program:* MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	After resolving the TT issue, the M&O operator queries for the appropriate TT, modifies the status (move to Closed) and submits the modification to the Remedy system.
Forward Trouble Ticket to another site	One per trouble ticket forwarded	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* Unix sendmail	The M&O staff member sends the TT to another site to be archived or for escalation to a review board for action.
Send Notification to originator	One per notification sent to originator	*Program:* MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	The originators receive notification of closure on their TT via e-mail sent using UNIX sendmail on the host where Remedy is running.

Table 4.9.4.5-1. Trouble Ticket Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Submit New TT via API	One per new TT submit via API	*Classes:* MsAcUsrProfile MsAcUsrProfileMgr_1_0 *Programs:* MsTtHTMLItems, MsTtServiceRequestor, MsTtManager (COTS, Remedy ARS)	*Process:* aradmin GUI	Alternatively, an ECS user can submit a TT via the custom web interface. This generates a new TT to begin the TT resolution process with a notification to the TT administrator of the new TT.
List open TTs	One per list open TTs	*Classes:* MsAcUsrProfile MsAcUsrProfileMgr_1_0 *Programs:* MsTtHTMLMenu, MsTtHTMLItems, MsTtManager (COTS, Remedy ARS)	*COTS:* Remedy ARS	The ECS user can also query the Remedy system through the custom web interface to obtain a list of active TTs that they have submitted and their current status.
TT forwarded from other site	One per TT forwarded from another site	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* Unixsendmail	An external site can forward a TT to the local Remedy system using a predefined TT format. for TT resolution purposes.
Submit TT from proper e-mail format	One per submit of TT in proper e-mail format	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* armail daemon	The armail daemon monitors a mailbox (arsystem) for new mail messages that conform to the TT mail exchange format. Upon receiving a valid message, a new TT is created that begins the TT resolution process with the TT administrator being notified of a new TT.
Modify Schemas, Filters, etc.	One per modification to schemas, filters, etc.	*Program:* MsTtManager (COTS, Remedy ARS)	*Process:* aradmin GUI	The TT administrator modifies schemas and screen layouts, escalation policies and filters via the Remedy supplied aradmin GUI.

4.9.4.6 Trouble Ticket Data Stores

Table 4.9.4.6-1 provides descriptions of the data stores shown in the Trouble Ticket architecture diagram. Also, descriptions are provided for the configuration files used by the Trouble Ticket CSC.

Table 4.9.4.6-1. Trouble Ticket Data Stores

Data Store	Type	Functionality
ARSystem	Database	This database is controlled by Remedy and stores the information from each schema in its own table. There is no clear mapping of schema to table. The Sybase table names are usually similar to T1, T2, T13, etc. Information includes: 1. Trouble Ticket detailed information 2. Contact Log detailed information 3. User information for Remedy users 4. Group information for roles within Remedy 5. Menus used by the GUIs
/etc/ar.conf	config file	Used by Remedy to determine binary locations and license information
$AR/bin/armaild.conf	config file	Used by the armaild to determine who to send error messages to.

4.9.5 MCI - Network Backup/Restore Component

4.9.5.1 Network Backup/Restore Functional Overview

The Legato vendor's Networker package provides a suite of integrated tools for backup and recovery, archival and retrieval, and hierarchical storage management. The product supports multi-platform networks, contains a motif-based GUI with on-line help, and supports concurrent device support for parallel backup and recovery using up to 16 storage devices. Authorized users can perform scheduled and ad-hoc backups, recoveries, and other data management services. Networker software consists of two parts: a client portion, which runs on the systems to be backed up, and a server portion, which is the system to which the backup devices are connected. The client portion sends the data to be backed up to the server portion, which writes the data out to disk.

4.9.5.2 Network Backup/Restore Context

Not Applicable.

4.9.5.3 Network Backup/Restore Architecture

Figure 4.9.5.3-1 is the Network Backup/Restore architecture diagram. The diagram shows the events sent to the Network Server of the Network Backup/Restore CSC and the events the Network Server of the Network Backup/Restore CSC sends to other processes (network clients).

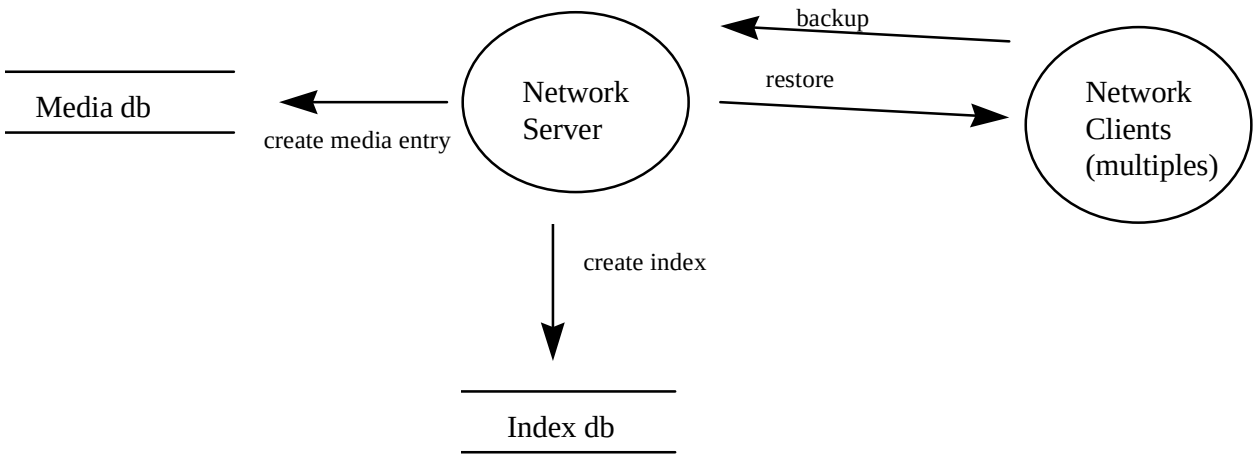

Figure 4.9.5.3-1. Backup/Restore Architecture Diagram

4.9.5.4 Network Backup/Restore Process Descriptions

Table 4.9.5.4-1 provides descriptions of the processes shown in the Backup/Restore architecture diagram.

Table 4.9.5.4-1. Backup/Restore Processes

Process	Type	COTS / Developed	Functionality
Networker Server	Server	COTS	The server can support multiple requester backups simultaneously. An index file is created to enable the backup operator to quickly find the proper tape from which to restore files or file systems.
Networker Client	Client	COTS	On each host that is backed up by Networker, a client portion is installed. The client portion can compress data before sending it to the server; however, doing so increases CPU usage on the client machine.

4.9.5.5 Network Backup/Restore Process Interface Descriptions

Table 4.9.5.5-1 provides descriptions of the interface events shown in the Backup/Restore architecture diagram.

Table 4.9.5.5-1. Backup/Restore Process Interface Events

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Backup	One per backup	COTS Client/ Server	*Process:* Networker Client	Data is passed from the client to the server and archived to tape.
Restore	One per restore	COTS Client/ Server	*Process:* Networker Server	Data is passed to the client to restore lost data from the tape backups.
Create Index	One per create index	COTS DB	*Process:* Networker Server	While saving data to tape, an index is created that gives the tape identification for any version of a file that needs to be restored.
Create media entry	One per create media entry	COTS DB	*Process:* Networker Server	After saving data files (save sets) to tape, the Networker Server makes an entry in the media db identifying what save sets are on the tape.

4.9.5.6 Network Backup/Restore Data Stores

Table 4.9.5.6-1 provides descriptions of the data stores shown in the Backup/Restore architecture diagram.

Table 4.9.5.6-1. Backup/Restore Data Stores

Data Store	Type	Functionality
Index db	Other	This proprietary index enables the backup operator to determine the location of the file(s) needing to be restored without searching all the tapes in the stacker. This index includes versioning information where appropriate.
Media db	Other	This media db tracks what file systems (save sets) are on each tape.

4.9.6 MCI - ASTER E-mail Handler Component

4.9.6.1 ASTER E-mail Header Handler Functional Overview

As specified in the Interface Between the ECS Communications and Systems Management Segment (CSMS) and the ASTER GDS CSMS Ground System Management Subsystem (GSMS) ICD (209-CD-002-005 8-1), a formatted header is added to all e-mail exchanges between the ASTER GDS and the ECS sites. The header contains information on the send date and time, the sender and receiver ID, and a unique output message sequence number. The header is detailed in 209-CD-002-005, page 8-6. Although the header is a necessary part of the ASTER to ECS e-mail transfer protocol, it does not contain information needed by ECS sending or receiving applications. The header therefore is automatically added to ECS e-mail sent to

ASTER and deleted from e-mail messages received from the ASTER GDS through the MSS provided ASTER e-mail header handler.

Using the ASTER to ECS e-mail transfer protocol, if a sequence number is skipped, the receiving site knows that a message has been lost and can request a retransmission. A log of messages sent and received through this header process is maintained by the ECS. The copies of messages are maintained in a format that enables the M&O staff to re-send a requested transmission using a standard UNIX mail tool such as Zmail.

Addition and deletion of the ASTER standard e-mail header is accomplished by creating aliases used by the Unix sendmail daemon. For instance:

A Trouble Ticket is to be sent to the ASTER GDS.

The Trouble Ticket is mailed to ECSTroubleTicket@<*edc.gov*>.

The sendmail daemon at <edc.gov> realizes that ECSTroubleTicket is an alias and filters the message through the AsterFilter.pl script.

The script adds the header information, logs and archives the message and forwards the message with header to the e-mail address specified in the alias, for example TroubleTicket@<aster.jp>.

A similar flow exists for the removal of the header when receiving e-mail from the ASTER GDS.

4.9.6.2 ASTER E-mail Header Handler Context

Figure 4.9.6.2-1 is the ASTER E-mail Header Handler context diagram. The diagram shows the events sent to the ASTER E-mail Header Handler and the events the ASTER E-mail Header Handler sends to ECS applications or the ASTER GDS. Table 4.9.6.2-1 provides descriptions of the events in the ASTER E-mail Header Handler context diagram.

Figure 4.9.6.2-1. ASTER E-mail Header Handler Context Diagram

Table 4.9.6.2-1. ASTER E-mail Header Handler Interface Events

Interface	Interface Event Description
ECS E-mail with header	An e-mail message (e.g., Expedited Data Set Request or EDR), containing an ASTER standard header, is sent from the ASTER GDS to a predefined e-mail alias at the ECS.
ECS E-mail without header	The header is removed from the inbound message, logged, and forwarded to the predefined ECS recipient of the e-mail alias.
ASTER E-mail without header	An e-mail message (e.g., Expedited Data Set Notification or EDN), without header, is sent by an ECS application to a predefined ASTER e-mail alias within the ECS.
ASTER E-mail with header	The header is added to the e-mail message by the e-mail handler and forwarded to the real ASTER destination.

4.9.6.3 ASTER E-mail Header Handler Architecture

Figure 4.9.6.3-1 is the ASTER E-mail Header Handler architecture diagram. The diagram shows the events sent to the ASTER E-mail Header Handler processes and the events the ASTER E-mail Header Handler processes send to the System Management Center and the ASTER GDS.

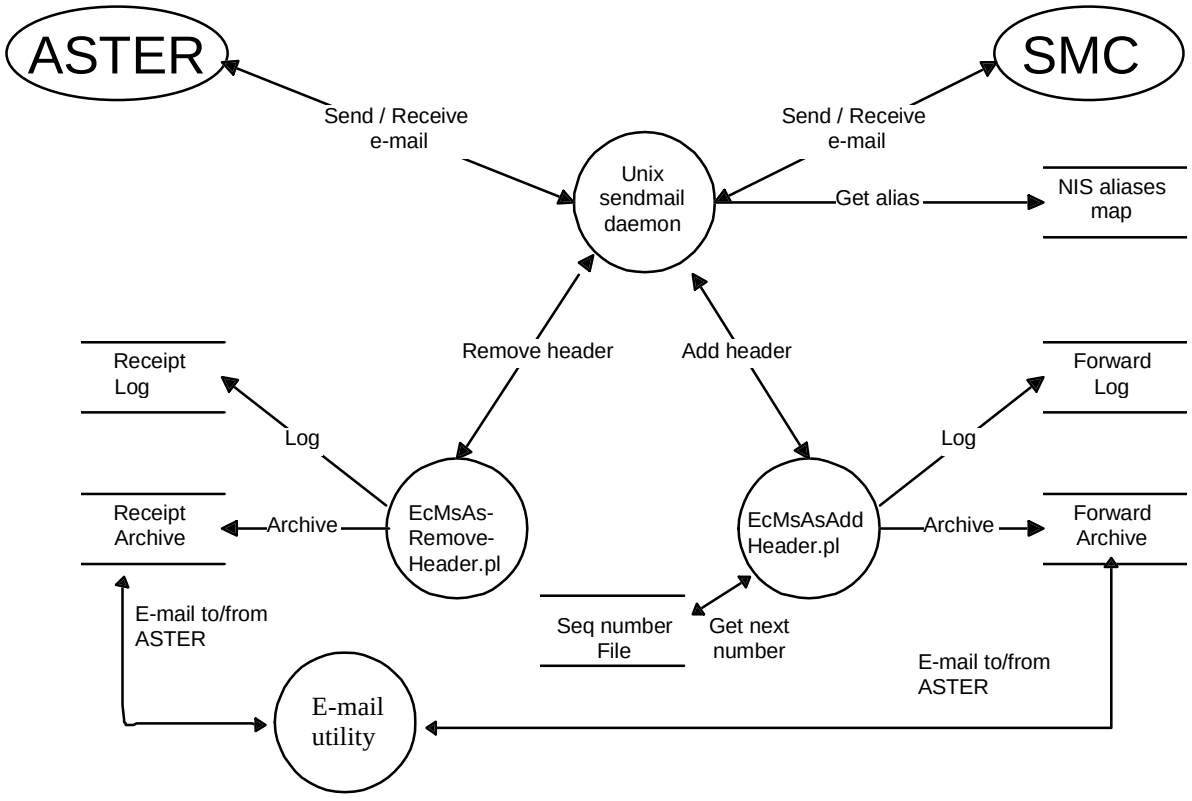

Figure 4.9.6.3-1. ASTER E-mail Header Handler Architecture Diagram

4.9.6.4 ASTER E-mail Header Handler Process Descriptions

Table 4.9.6.4-1 provides descriptions of the ASTER E-mail Header Handler processes shown in the ASTER E-mail Header Handler architecture diagram.

Table 4.9.6.4-1. ASTER E-mail Header Handler Processes

Process	Type	COTS / Developed	Functionality
Unix sendmail daemon	Server	COTS	The sendmail daemon, which is provided with the UNIX operating system, handles delivery and receipt of e-mail messages.
EcMsAsAddHeader.pl	Other	Developed	This script is invoked by the sendmail daemon when a message is sent to an alias configured to process messages requiring ASTER e-mail headers before delivery. This perl script inserts the header into a message directed to the ASTER GDS.
EcMsAsRemoveHeader.pl	Other	Developed	This script is also invoked by the sendmail daemon when a message is sent to an alias configured to process e-mail containing ASTER e-mail headers. The perl script removes the header and forwards the message to the address defined in the alias.
E-mail utility	Other	COTS	An e-mail utility (such as Zmail) is used by the M&O staff for review of transmitted/received messages and the messages can be retransmitted in the event of a problem.

4.9.6.5 ASTER E-mail Header Handler Process Interface Descriptions

Table 4.9.6.5-1 provides descriptions of the interface events shown in the ASTER E-mail Header Handler architecture diagram.

Table 4.9.6.5-1. ASTER E-mail Header Handler Process Interface Events (1 of 2)

Interface	Event Frequency	Interface	Initiated By	Interface Event Description
Send e-mail	One per e-mail send	API, system call or command line	Unix Sendmail daemon	The unix sendmail daemon attempts to forward e-mail messages to the specified recipient.
Receive e-mail	One per e-mail receive	SMTP protocols	Unix Sendmail daemon	The unix sendmail daemon receives and processes e-mail messages it has been configured to receive.

Table 4.9.6.5-1. ASTER E-mail Header Handler Process Interface Events (2 of 2)

Interface	Event Frequency	Interface	Initiated By	Interface Event Description
Get alias	One per get alias	NIS system call	Unix Sendmail daemon	While processing e-mail, the sendmail daemon checks to see if the specified recipient is a local user, an alias for another user, or an executable to stream the message into.
Add header	One per header added	Perl interpreter	EcMsAsAddHeader.pl (script)	The ASTER e-mail header is inserted in the body of a e-mail message by the EcMsAsAddHeader.pl script.
Remove header	One per header removed	Perl interpreter	EcMsAsRemoveHeader.pl (script)	The ASTER header is removed from messages sent for local delivery by the EcMsAsRemoveHeader.pl script.
Log	One per log	Perl interpreter	EcMsAsAddHeader.pl (script) EcMsAsRemoveHeader.pl (script)	An entry is added to note the e-mail message date, time, recipient, etc. being forwarded.
Archive	One per archive	Perl interpreter	EcMsAsAddHeader.pl (script) EcMsAsRemoveHeader.pl (script)	A copy of the e-mail message is stored in a format that can be read by Z-mail (Setenv MAIL to the file location of the archive)
Get next number	One per get of next number	Perl interpreter	EcMsAsAddHeader.pl (script)	The EcMsAcAddHeader.pl script obtains the next sequence number from a text file.
E-mail to/from ASTER	One per e-mail to/from ASTER	SMTP protocols	E-mail utility (e.g., Zmail)	E-mail messages with standard headers are sent to/from ECS users or M&O staff personnel from/to ASTER GDS users or operations personnel.

4.9.6.6 ASTER E-mail Header Handler Data Stores

Table 4.9.6.6-1 provides descriptions of the data stores shown in the ASTER E-mail Header Handler architecture diagram.

Table 4.9.6.6-1. ASTER E-mail Header Handler Data Stores

Data Store	Type	Functionality
NIS database	Other	This database provides the aliases used by sendmail to determine where to redirect the e-mail messages.
Seq number file	text file	This file contains the next available sequence number.
Receipt Log	text file	The date/time stamp and recipient are maintained in this log.
Forward Log	text file	The date/time stamp and recipient are maintained in this log.
Receipt Archive	text file	This file maintains copies of e-mail messages sent from the ASTER GDS.
Forward Archive	text file	This file contains copies of e-mail messages sent to the ASTER GDS.

4.9.7 MACI Software Description

The MACI CSCI is the following ECS developed and COTS software: SubAgent, Deputy Agent, Proxy Agent, and Master Agent.

The ECS subagent communicates management requests and responses from the master agent and the Deputy agent to either an ECS developed application or to a COTS application via the Proxy Agent. It supports the MIB extensions and performs local polling on the resources on the host. The subagent is custom developed software using some COTS libraries such as PEER tools to make itself remotely manageable.

4.9.7.1 MACI Functional Overview

The Management Agent (MA) CI manages and monitors ECS applications (via the CSS process framework managed servers and COTS applications). The Deputy agent handles secure delivery of requests for setting management information by using DCE remote procedure calls. The Proxy agent manages non-SNMP manageable COTS products. Its front-end has the MSS instrumentation software to communicate with the subagent. Its back-end communicates with the COTS. Each COTS process provides a definition to the Proxy agents, which includes its start up and shutdown procedure description. The Proxy agent instantiates a manager object on behalf of each COTS process started. This manager object binds with and is monitored by the subagent. The Master agent is an SNMP agent that manages resource distribution to one or more subagents using a client/server communications paradigm. The communication between the client and the server is encapsulated in the SNMP Multiplexing (SMUX) protocol. When a sub-agent is started, it connects with the master agent running on the host. If the connection is successful, the subagent registers the branch of the MIB it is managing with the master agent.

The Encapsulator is a specialized sub-agent that enables incompatible and non-extensible SNMP agents to be present on the same processor as the Master Agent. These SNMP agents are vendor supplied agents installed on the workstation as part of the Operating System.

It is assumed that the master agent and the subagent, responsible for the management of applications, are always running. They are started when the host is booted. In addition, Tivoli monitors the status of the subagent and attempts to restart it if it is not running. The request to

start or shutdown an application can be issued on the management application. This request is passed securely to the subagent on remote hosts and the startup or shutdown actions are performed. The life cycle services (i.e., startup, shutdown, and other requests) trigger event notifications to the MSS enterprise and management framework. These requests are bundled within RPCs by the HP OpenView custom software and sent to the remote subagent's Deputy Gate.

The MSS requires each managed host, standard MIB, Host Resource MIB, and the network device MIBs to be supported by vendor agents. In addition, a managed object model is defined by the MSS for ECS applications in the SNMP MIB format as the ECS application MIB. The management agent service implements the ECS application MIB. The MIB information is composed of different types of attributes: configuration, performance, fault, dynamic, static, and traps.

Management applications can make SNMP requests to retrieve management information as MIB values. They can also set certain management information via secure DCE RPCs.

4.9.7.2 MACI Context

Figure 4.9.7.2-1 is the MACI CSCI context diagram. The diagram shows the events sent to the MACI CSCI and the events the MACI CSCI sends to the HP OpenView COTS product. Table 4.9.7.2-1 provides descriptions of the interface events shown in the MACI CSCI context diagram.

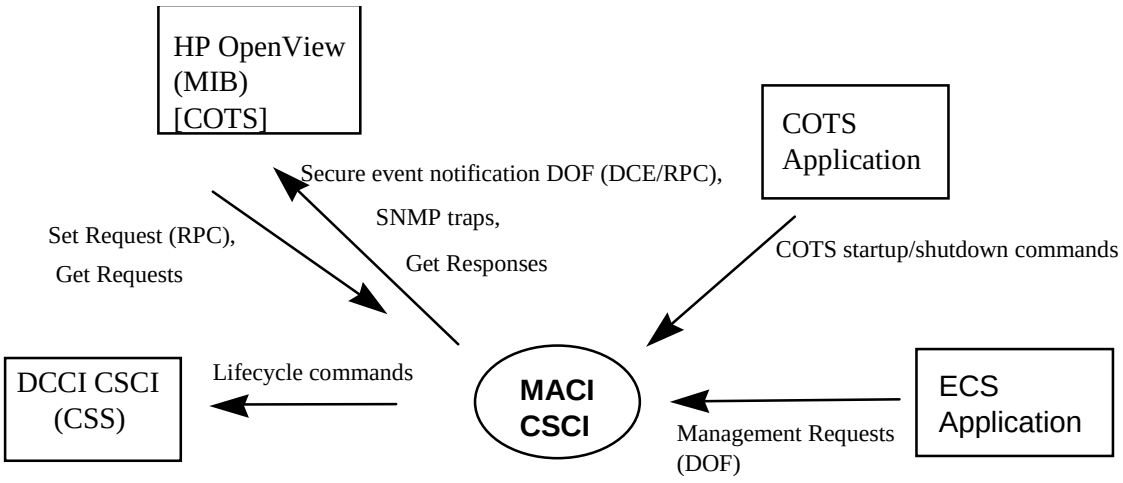

Figure 4.9.7.2-1. MACI CSCI Context Diagram

Table 4.9.7.2-1. MACI CSCI Interface Events

Event	Interface Event Description
Management Requests (DOF)	When an ECS application is started, it sends a request to the Registry service in the Subagent to start monitoring it.
SNMP traps	The Deputy Agent converts the application events received from the subagent and converts the events into traps. These traps are forwarded to the trapd daemon. Trapd logs the traps into the trapd log where they are read by HP OpenView (HPOV). For example, if an application dies, the subagent sends a topology change event to the Deputy Agent. The Deputy Agent converts the change event into a trap and forwards the trap, the trap is read by HPOV, and the corresponding icon on the GUI turns red to indicate the application died.
Get Requests (SNMP)	The HP OpenView management platform sends the Get requests to the Peer Master Agent running on theMSS server or ECS host. The SNMP protocol is used to send the requests.
Get Responses (SNMP)	The Peer Master Agent returns the responses for Get requests from the Subagent to theHP OpenView management platform.
COTS startup/shutdown commands	The Proxy Agent sends the startup and shutdown requests to the COTS applications. The startup and shutdown script associated with each COTS product is listed in the Proxy rules file (MsAgGenProxy.XCFG). Note: The Tivoli COTS product can monitor COTS applications and restart them if they are not running.
lifecycle commands	HP OpenView issues startup/shutdown lifecycle commands via the MSS management sub-agent to applications in the managed hosts.
application events	Errors logged by ECS applications are sent to the ALOG file. Application events can also be generated by the subagent in response to topology changes such as a process startup/shutdown.

4.9.7.3 MACI Architecture

Figure 4.9.7.3-1 is the MACI CSCI architecture diagram. The diagram shows the events sent to the MACI CSCI management agent processes and the events the MACI CSCI management agent processes send to each other, other CSCIs and COTS products.

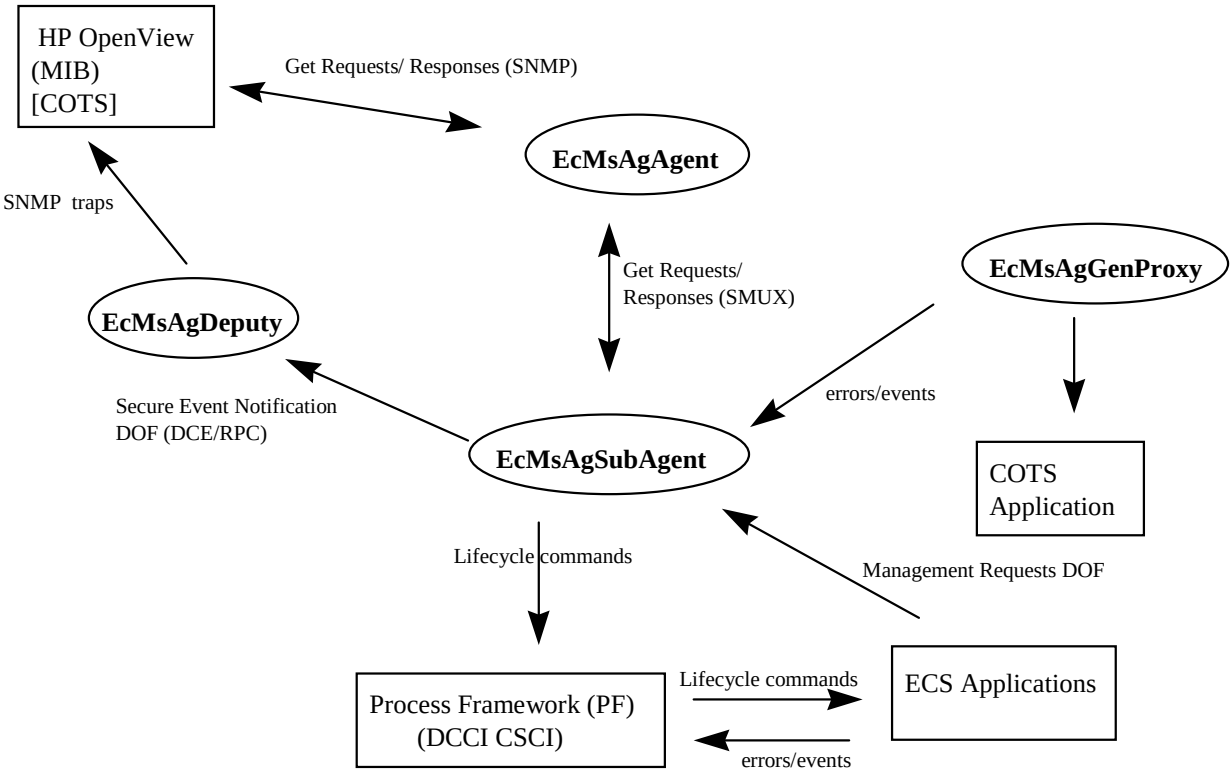

Figure 4.9.7.3-1. MACI CSCI Architecture Diagram

4.9.7.4 MACI Process Descriptions

Table 4.9.7.4-1 provides descriptions of the processes shown in the MACI CSCI architecture diagram.

Table 4.9.7.4-1. MACI CSCI Processes

Process	Type	COTS / Developed	Functionality
EcMsAgSubAgent	Other	COTS/Developed	The EcMsAgSubAgent manages startup, shutdown, and monitoring of ECS developed and COTS applications.
EcMsAgDeputy	Other	COTS/Developed	The Deputy Agent receives application events from the EcMsAgSubAgent in the CDS cell, converts the application events into traps and forwards the traps to the HP Open View COTS package.
EcMsAgAgent	Other	COTS	The Master Agent is an SNMP agent that manages the distribution of application management requests to the EcMsAgSubAgent on each host.
EcMsAgGenProxy	Other	Developed	The Proxy Agent manages non-SNMP COTS products.

4.9.7.5 MACI Process Interface Descriptions

Table 4.9.7.5-1 provides descriptions of the interface events shown in the MACI CSCI architecture diagram.

Table 4.9.7.5-1. MACI CSCI Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Manage-ment Requests (DOF)	One per managemen t request	*Classes:* MsAgRegistry MsAgEventHandler	*Processes:* ECS Applications	When an ECS application is started, the application sends a request to the Registry service in the EcMsAgSubAgent to start monitoring it.
Get Requests / Responses (SMUX)	One per get of requests/res ponses	*Classes:* MsAgPortMonitor *Structure:* MsAgAppMIB	*Processes:* EcMsAgAgent (request) EcMsAgSubAgent (responses)	The EcMsAgSubAgent connects to the EcMsAgAgent when it starts. If successful, the EcMsAgSubAgent registers the root of the ECS MIB with the EcMsAgAgent, and begins to monitor for ECS MIB requests. If not successful, the EcMsAgSubAgent periodically tries to connect to the EcMsAgAgent and log error messages indicating the connection failure until a successful connection takes place. All SNMP Get Requests intended for receipt by the ECS MIB are sent to the EcMsAgSubAgent by the EcMsAgAgent, and responses are returned from the EcMsAgSubAgent to the EcMsAgAgent which formulates an SNMP response to the management platform.

Table 4.9.7.5-1. MACI CSCI Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
SNMP traps	One per SNMP trap	*Classes:* MsAgDeputyGate MsAgSnmpPdu	*Process:* EcMsAgDeputy	The EcMsAgDeputy converts the application events received from the EcMsAgSubAgent and converts them into traps. The traps are forwarded to the trapd daemon. Trapd logs the traps into the trapd log where they are read by HPOV. For example, if an application dies, the EcMsAgSubAgent sends a topology change event to the Deputy Agent. The Deputy Agent converts the change event into a trap and forwards the trap that is read by HPOV. HPOV causes the corresponding icon on the GUI to turn red to indicate the application has died.
Secure Event Notification DOF (DCE/RPC)	One per secure event notification	*Classes:* MsAgEventMgr EcAgEvent	*Process:* EcMsAgSubAgent	All the events from the EcMsAgSubAgent are sent to a Deputy Agent via a RPC.
Get Requests and Responses (SNMP)	One per get request and response	*Class:* MsAgAgent (COTS)	*Processes:* Hp OpenView (request) EcMsAgAgent (response)	The HP OpenView management platform sends the Get requests to the EcMsAgAgent running on the remote host. The SNMP protocol is used to send the get requests.
COTS startup/shut down commands	One per startup or shutdown command	*Class:* MsAgGenProxy	*Process:* EcMsAgGenProxy	The EcMsAgGenProxy agent sends the startup and shutdown commands to the COTS applications. The startup and shutdown script associated with each COTS is listed in the Proxy rules file (MsAgGenProxy.XCFG).
Lifecycle Commands	One per lifecycle command	*Class:* MsAgDeputyGate	*Process:* EcMsAgSubAgent	HP OpenView sends startup/shutdown lifecycle commands to the EcMsAgSubAgent, via the CSS Process Framework Library calls, which are sent to applications installed on the managed hosts.

Table 4.9.7.5-1. MACI CSCI Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Interface Event Description
Application Events	One per application event	*Classes:* EcAgEvent MsAgPerfEvent MsAgEventMgr	*Process:* CSS (PF)	Includes events contained in the system log (such as too many logins).

4.9.7.6 MACI Data Stores

Not applicable.

4.9.8 MLCI - Baseline Manager Component

4.9.8.1 Baseline Manager Functional Overview

Baseline Manager aids the DAACs, EOC, and SMC staffs in maintaining records to describe the base-lined, operational system configurations. These records primarily identify the versions of hardware and software items the baselines contain. These records also identify item interdependencies and the sites where the items are deployed. Additionally, the records also track the identity of devices, subsystems, and networks, maintaining historical change records and traceability of version-controlled items to their predecessors and associated system releases. Baseline Manager at the DAACs and EOC maintains records about baselines deployed at the site. At the SMC, Baseline Manager maintains records about baselines system-wide. A COTS application called XRP II is used to accomplish baseline management tasks.

The functionality for the Baseline Manager is implemented via the COTS products XRP-II and ACCELL. The following are the only custom files for the Baseline Manager:

- README.xrp (text file) - provides instructions for installing and configuring the XRP-II application.
- README.accell (text file) - provides instructions for installing and configuring the COTS product ACCELL and its UNIFY RDBMS which is used by XRP-II.
- scr_perm.doc (Microsoft (MS) Word file) - contains a table describing inquire, add, modify, and delete permissions assigned by default to screens for XRP-II user groups established for ECS.

4.9.8.2 Baseline Manager Context Diagram

Baseline Manager runs at the SMC, EOC, and each DAAC. Baseline records can be exchanged among sites via formatted data files. These files are created locally on demand and transferred to other sites as appropriate via the ftp service where Operations staff uses them to update their site’s database.

Baseline Manager has one interaction with another MSS CSCI, namely MCI. As shown in Figure 4.9.8.2-1, the Baseline Manager provides select baseline data records and an associated end-of-task signal to MCI (Tivoli) in response to Resource Planner (resplan) data requests.

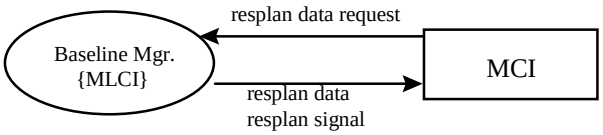

Figure 4.9.8.2-1. Baseline Manager Context Diagram

4.9.8.3 Baseline Manager Architecture

Baseline Manager is implemented as a specially configured version of the commercially available manufacturing management system "XRP-II". It is a single, standalone, non-client/server application with an internal relational database. XRP-II uses the UNIFY relational database management system marketed as part of the ACCELL Integrated Developmentenvironment product. The database (refer to Figure 4.9.8.3-1) is shared with the Inventory/Logistics/Maintenance (ILM) Manager CSC (which is also implemented using XRP-II). Data records can be exported via formatted data files for use by Baseline Manager applications at other sites.

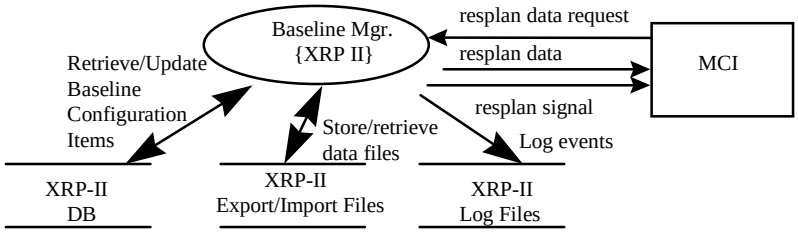

Figure 4.9.8.3-1. Baseline Manager Architecture Diagram

Baseline Manager's controlling program is the menu handler, "xrp". Invoked via the startup script "pcs", the menu handler uses the Operations staff member userid to present a character-based user interface with menus, data entry screens, and permissions for functions and menus the Operations staff are authorized to use.

4.9.8.4 Baseline Manager Process Descriptions

The XRP-II menus and screens invoke sub-processes that perform functions summarized in Table 4.9.8.4-1.

Table 4.9.8.4-1. Baseline Manager Processes

Process	Type	COTS / Developed	Functionality
Baseline Mgr (XRP-II)	Other	COTS	• Manages records identifying deployed ECS baselines and the versions of hardware and software items the baselines contain. • Maintains chronological histories of baseline changes. • Maintains item traceability to predecessors and associated system releases. • Exports/imports records using formatted data files to support data exchange between the DAACs and the SMC. • Generates both pre-defined and ad hoc reports about baseline-related data

4.9.8.5 Baseline Manager Process Interface Descriptions

Baseline Manager's sole ECS processing interface is with the MCI CSCI's Tivoli application. When commanded by resource planners, Tivoli issues Baseline Manager a request for resource planner data. Tivoli makes the request by running the Baseline Manager's "resplan" script, passing a date and code as arguments. Baseline Manager responds by producing a set of formatted data records and a signal containing a status message. The data records are written to stdout, which Tivoli reads and routes to a predetermined path name and host as part of Tivoli's configuration. The signal, sent to Tivoli via its "wasync" command, passes an action code and a message which Tivoli can use for notifying Resource Planners (e.g., in a pop-up window) that the data records have been delivered. Table 4.9.8.5-1 provides descriptions of the interface interface events shown in the Baseline Manager architecture diagram.

Table 4.9.8.5-1. Baseline Manager Process Interface Events (1 of 2)

Event	Interface Event Description
resplan data request	Request to XRP-II for data about hosts, hardware, software, and disk partitions constituting the site's production baseline as of a specified date. Arguments associated with the request are the baseline date and a code used by MCI to notify Resource Planners of the outcome of the request. The request format is: resplan <mmddyy> <code>
Resplan Signal	Notification for MCI that the resplan data request has been processed. The notice is made via Tivoli's "wasync" utility. It contains: • a code indicating the purpose of the notice • an associated informational message.
Retrieve/Update Baseline Configuration Items	Maintains records of hardware and software configuration items for ECS. Tracks the identity of all devices and software items. Maintains a change history for all baseline configuration items.

Table 4.9.8.5-1. Baseline Manager Process Interface Events (2 of 2)

Event	Interface Event Description
Store/Retrieve data files	Store data that is imported from other sites (DAACs, DOC, or SMC), to be recorded at the receiving site. Retrieves data that needs to be sent to other sites for recording as a backup.
Resplan Data	Set of ASCII records containing one header record followed by one or more detail records. The header record contains a text message identifying the production baseline that was specified and the number of data records. Detail records describe the items marked as "planning resource" in the Baseline Manager database that constitute the site's production baseline as of an operator-specified date. Data in a detailed record is separated by a pipe symbol "|" and varies by type of item as follows: • host items - "host", name, description, control item id, status, install date, # CPUs, total RAM, processing string name, string's control item id • hardware items - "hardware", name, description, control item id, status, install date • software items - "software", description, version, control item id, status, install date, associated host name, host control item id • disk partition items - "partition", device name, directory name, control item id, status, install date, partition size, block size, logical allocation, associated host's name, host's control item id • processing string items - "string", control item id, name, description, status, install date
Log events	Stores records describing notable events as described in Table 4.9.8.6-1 under data store named "XRP-II log files"..

4.9.8.6 Baseline Manager Data Stores

Baseline Manager's principal data stores are the XRP-II database, log files, and formatted data files used for exporting and importing Baseline Manager records. The data store descriptions are provided in Table 4.9.8.6-1.

Table 4.9.8.6-1. Baseline Manager Data Stores (1 of 2)

Data Store	Type	Description
XRP-II DB	database	A non-replicated collection of baseline, inventory, and maintenance-related data that exists at each site. For Baseline Manager, it principally contains records identifying and describing: • control items: version-controlled entities such as baselines, software products, hardware devices, and documents • product structures: parent/product component pairings defining the ingredients – or bill of material – for control item assemblies • engineering change notices: mechanisms by which configuration changes with their effectivity dates are defined for an assembly • control item implementation status – mappings of control items against deployment sites, together with status and date for each • control item interdependencies – specified dependencies that any control item has on another.

Table 4.9.8.6-1. Baseline Manager Data Stores (2 of 2)

Data Store	Type	Description
XRP-II export/import files	tar file	Formatted data files created as necessary to exchange Baseline Manager records among sites. Each contains either: • all site-unique Baseline Manager records new or changed at a site since the previous export of changed site-unique records • all "core" Baseline Manager records changed since the previous export of changed "core" records • all the Baseline Manager records for a specified system release, baseline, or other configuration-controlled item or assembly • all records dumped from one or more XRP-II database tables
XRP-II log files	text files	A collection of files containing information about XRP-II events and errors encountered during processing including: • xrp.log - userid, date/time, and result of operator attempts to log into XRP-II • datadump.log - userid, date/time, and result of operator attempts to dump XRP-II data in bulk into ASCII files • dataread.log - used to load XRP-II data in bulk from ASCII files • errlog and *.err files - details about fatal errors; useful mainly to XRP-II programmers • import.log - events associated with importing data from other sites.
UNIFY Transaction log	binary file	A collection of records for journaling and rolling forward database transactions. Used in conjunction with database backups and restores.

4.9.9 MLCI - Inventory/Logistics/Maintenance Manager Component

4.9.9.1 Inventory/Logistics/Maintenance Manager Functional Overview

The Inventory/Logistics/Maintenance (ILM) Manager tracks and maintains all of the key data pertaining to ECS contract purchased equipment including hardware, COTS software, COTS documentation (hardware and software), spares and consumable items, and Government Furnished Equipment (GFE). The type of information includes date of receipt, installation, and warranty expiration, user, location, manufacturer, vendor, Original Equipment Manufacturer (OEM) part number, model version, and description. The ILM Manager also stores and maintains detailed maintenance data on hardware, to the item level, including preventive and corrective maintenance.

The ILM Manager gives users rapid access to property data via a character-based interface. The user can select individual records, or subsets of data, by entering one or more valid attributes in a data entry screen. Reporting capabilities are available to the user by accessing the reports menu and entering valid values. The user can also obtain ad-hoc reports, from any screen within the ILM Manager, by entering "/R" at the command prompt. The ILM Manager enables users to select from three formats and three destinations for their output including table based reports, form based reports and ASCII output to screen, files, or printers.

For maintenance of contract hardware and COTS software, the ILM Manager tracks the OEM warranty expiration dates, maintenance contract, phone numbers, contact, maintenance password and license key data. The COTS application called XRP II used for baseline management is also used to accomplish the Inventory, Logistics, and Maintenance functions.

The functionality for the Inventory/Logistics/Maintenance Manager Computer Software Component (CSC) is implemented via the COTS products XRP-II and ACCELL. No custom scripts are used in this CSC. The following is the only custom file for the CSC.

- README.xrp (text file) - provides instructions for installing and configuring the Baseline Manager and Inventory/Logistics/Maintenance Manager CSCs.

4.9.9.2 Inventory/Logistics/Maintenance Manager Context Diagram

The ILM Manager does not have an interface with any other subsystem CSCIs.

4.9.9.3 Inventory/Logistics/Maintenance Manager Architecture

The ILM Manager is implemented as a specially configured version of the commercially available manufacturing management system "XRP-II". It is a single, standalone, non-client/server application with an internal relational database. XRP-II uses the UNIFY relational database management system marketed as part of the ACCELL Integrated Development System. The database is shared with the Baseline Manager CSC. Data records can be exported via formatted data files for use by the ILM Manager applications at other sites. Figure 4.9.9.3-1is the ILM Manager architecture diagram.

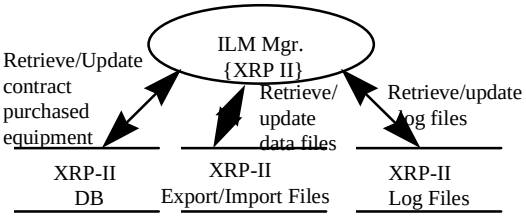

Figure 4.9.9.3-1. ILM Manager Architecture Diagram

4.9.9.4 Inventory/Logistics/Maintenance Manager Process Descriptions

ILM's controlling program is its menu handler, xrp. XRP-II is invoked via the startup script "ilmusr". The menu handler uses the operator's userid to present a character-based interface with menus, data entry screens, and permissions for functions and menus the operators are authorized to use. The menus and screens in turn invoke sub-processes that perform functions summarized in Table 4.9.9.4-1.

Table 4.9.9.4-1. ILM Manager Processes

Process	Type	COTS / Developed	Functionality
ILM Mgr. (XRP-II)	Other	COTS	The ILM performs the following tasks: • Captures and maintains all pertinent data for project hardware and COTS software. • Manages, distributes, maintains reorder thresholds, and reports on consumables and spares. • Manages, controls and reports on preventative maintenance actions. • Manages, controls and reports on maintenance items other than preventative maintenance actions. • Maintains historical log of maintenance actions against individual items within ILM. • Tracks movement and archive actions for project property and reports on same. • Manages and controls receipts against purchase orders • Maintains all other pertinent property management data required for the efficient use of the ILM tool such as vendor, buyer, user, manufacturer, and internal usage codes.

4.9.9.5 Inventory/Logistics/Maintenance Manager Process Interfaces

Table 4.9.9.5-1 provides descriptions of the interface events shown in the ILM Manager architecture diagram.

Table 4.9.9.5-1. ILM Manager Process Interface Events

Event	Interface Event Description
Retrieve/Update contract purchased items	Maintains and retrieves contract purchased equipment information for ECS hardware, COTS software, COTS documentation, spare parts, consumable items such as printer ribbons and GFE. Tracks warrantees, licenses, and maintenance for equipment.
Retrieve/Update data files	Maintains data files of contract purchased equipment from other sites (DAACs, EOC, SMC) and sends information to other sites for spare and consumable parts information as well as for backup/retrieve purposes.
Retrieve/Update log files	Logs information on activities performed by this COTS package. Retrieves and sends log files to the SMC for long term archival.

4.9.9.6 Inventory/Logistics/Maintenance Manager Data Stores

ILM Manager's principal data stores are the XRP-II database and formatted data files used for exporting and importing ILM records. The data store descriptions are provided in Table 4.9.9.6-1.

Table 4.9.9.6-1. ILM Manager Data Stores

Data Store	Type	Functionality
XRP-II DB	database	A non-replicated collection of baseline, inventory, and maintenance-related data that exists at each site. For ILM, it contains records identifying and describing: • inventory items • maintenance schedules • location data • user data • maintenance support information
XRP-II export/import files	tar file	Formatted data files created as necessary to exchange the ILM Manager records among sites. Each file contains: • all changed records pertaining to receipts, installations, archives, transfers, relocations, shipments, manual changes, or maintenance that is performed regardless of the site.
Log files	Text Files	A collection of files containing information about XRP-II logon attempts and errors encountered during processing as follows: • xrp.log - userid, date/time, and result of operator attempts to log into XRP-II • errlog and *.err files - details about fatal errors; useful mainly to XRP-II programmers

4.9.10 MLCI - Software Change Manager Component

4.9.10.1 Software Change Manager Functional Overview

The Software Change Manager aids the DAACs, EOC, and SMC staffs in organizing and partitioning software, controllng software changes and versions, and in assembling sets of software for release purposes. The Software Change Manager consists of a COTS application called ClearCase.

4.9.10.2 Software Change Manager Context Diagram

The Software Change Manager does not interact with any CSCIs or CSCs.

4.9.10.3 Software Change Manager Architecture

The Software Change Manager (ClearCase) does not interface with any external processes.

4.9.10.4 Software Change Manager Process Descriptions

The Software Change Manager's primary process is the COTS package, ClearCase. ClearCase has both a command line and a graphical user interface to execute its programs. Table 4.9.10.4-1 provides a summary of its functions.

Table 4.9.10.4-1. Software Change Manager Processes

Process	Type	COTS / Developed	Functionality
SW Change Mgr. (ClearCase)	Other	COTS with custom developed scripts	• Organizes and stores software in a software library. • Manages multiple versions of software files. • Regulates access to software file versions. • Controls and logs changes to software file versions. • Manages software file version's progress through the development cycle. • Performs builds of software according to user defined version specifications. Maintains records of a build's content (files, compiler, and other resources used).

4.9.10.5 Software Change Manager Process Interface Descriptions

Not Applicable.

4.9.10.6 Software Change Manager Data Stores

The Software Change Manager's COTS package, ClearCase's data stores consist of a database and log files. Table 4.9.10.6-1 provides descriptions of the data stores shown in the Software Change Manager architecture diagram.

Table 4.9.10.6-1. Software Change Manager Data Stores

Data Store	Type	Description
ClearCase Database	Database	ClearCase uses a proprietary database management/database scheme that consists of versioned object base(s) (VOB) and views. A VOB is a data structure mounted as a multi-version file system and is created through use of the ClearCase "make vob" command. A VOB contains versions of directories and files, user-defined metadata, build records, event records, and configuration records. A view is also a data structure that's used as short-term storage for data created during the development process. A view stores checked-out versions of file elements, a user's private files, and newly built derived objects.
ClearCase Log Files	Other	ClearCase log files record error and status information from various ClearCase server programs and user programs. These log files are ASCII files and are described in the ClearCase Reference Manual.

4.9.11 MLCI - Change Request Manager Component

4.9.11.1 Change Request Manager Functional Overview

The Change Request Manager enables the DAACs, EOC, and SMC staffs to enter, maintain, and keep track of configuration change requests (CCRs) electronically. A COTS application called the DDTS, (Version 3.2, with some customizations) is used to perform the change request functions. Each site (System Management Center (SMC), Distributed Active Archive Centers (DAACs), Earth Observing System Operations Center (EOC)) has a copy of the Change Request Manager. This gives the sites the capability to compose and maintain local CCRs and also compose and submit ECS CCRs to the SMC for system-wide consideration. Communications between site Change Request Managers can be established via a DDTS utility program and maintained by each site's DDTS administrator.

Description: The Change Request Manager (CRM) provides the functionality to compose, submit, coordinate, and track the status of CCRs. DDTS, a COTS application, forms the basis of the Change Request Manager and it has been customized to include a configuration change request form for inputting CCR information. DDTS records are organized into DDTS classes (note, this is not referring to object classes) and projects. Each DDTS class has its own process model and rules for how records are handled. Each DDTS class consists of one or more projects. A project is used to group DDTS records for a specific development product. The customization of DDTS and explanations of its terminology, directories, files, and database are covered in the DDTS Administrator's Manual.

Contents: The Change Request Manager CSC contains the following custom files.

- Readme File: This file describes how to install the customized DDTS.
- Change_Request Class: The Change_Request Class was developed to handle CCR information. This is a custom DDTS class added to the ~ddts/class directory that enables the operations staff to enter configuration change request information into the DDTS database. This class was developed through the use of adminbug, a DDTS utility program that creates new classes through the use of information in the customizing section of the DDTS Administrator's Manual. Therefore, Change_Request contains the standard set of DDTS class directories and files and uses standard DDTS code formats. To implement the Change_Request class, changes were made to the following class directory elements:
    - master.tmpl: Change_Request class directory file contains the DDTS code executed when the Change_Request class is selected for use. It defines the rules for moving a CCR from state-to-state by enabling interactive dialogue and requesting the data for state transition, it defines how the fields for a CCR are displayed on the monitor or printer, and it defines how a CCR is formatted for e-mail.
    - oneofs directory : The "oneofs" directory contains a set of files where each file defines a list of valid responses for an associated field. "oneofs" files added include: CCB-Organization, Change-Class, Disposition, Eval-Organization, Impact-Evaluators, Impl-Organization, Priority, Sites-Affected.

- helps directory: The "helps" directory contains a set of files where each file provides descriptive information about an associated field. A help file was developed for each new field added to the database.
- The following CCR related fields were added to the DDTS database:

baselines_affected	ccb_approval_date	ccb_approval_official
ccb_organization	change_class	ci_affected
closed_by	closing_date	closing_org
completion_date	disposition	docs_affected
effective_date	est_time_to_complete	eval_organization
impact_eval(1 - 12)	impacts_project	impl_engr
impl_engr_email	impl_organization	manager
originator_name	originator_org	originator_phone
originator_eval_engr	priority_string	related_ccr
release_affected	site_affected (1 - 9)	start_date
submitter_host	submitter_org	submitter_phone
test_est_complete_date	test_org	test_status
verify_engineer		

4.9.11.2 Change Request Manager Context Diagram

The Change Request Manager does not interact with any other CSCIs or CSCs.

4.9.11.3 Change Request Manager Architecture

The Change Request Manager (DDTS) does not interface with any other processes.

4.9.11.4 Change Request Manager Process Descriptions

The Change Request Manager's primary process is the COTS package, DDTS. DDTS has both a character based and a graphical user interface. Table 4.9.11.4-1 provides descriptions of the processes shown in the Change Request Manager architecture diagram.

Table 4.9.11.4-1. Change Request Manager Processes

Process	Type	COTS / Developed	Functionality
Change Request Manager (DDTS)	Other	COTS	• Facilitates the entry, update, and recording of CCR and NCR information. • Organizes and stores CCRs and NCRs in a database. • Maintains the status and disposition of CCRs and NCRs as they advance through the approval and implementation processes. • Provides reports of CCR and NCR information.

4.9.11.5 Change Request Manager Process Interface Descriptions

Not Applicable

4.9.11.6 Change Request Manager Data Stores

The Change Request Manager's data stores consist of a database and a log file. Table 4.9.11.6-1 provides descriptions of the data stores shown in the Change Request Manager architecture diagram.

Table 4.9.11.6-1. Change Request Manager Data Stores

Data Store	Type	Functionality
DDTS Log File	Other	The DDTS log file records CCR/NCR update activity, status information and error messages from various DDTS programs.
DDTS DB	Database	DDTS has a proprietary database management/database schema and the database is described in the DDTS Administrator's Manual. The database consists of three tables: defects table (stores all of the basic CCR and NCR information), enclosures table (stores CCR/NCR related files), and change_history table (stores the complete history for each CCR and NCR as it progresses through its life cycle). Usually, there is one database per site. The DDTS procedures facilitate partial replication of a site's database at other sites.

4.9.12 MLCI - Software Distribution Manager Component

4.9.12.1 Software Distribution Manager Functional Overview

The Software Distribution Manager enables the SMC and the DAAC staffs to distribute ECS software, database, software documentation, and commercial software files across a multi-platform ECS network. A COTS application called Tivoli/Courier is used to perform the software distribution functions. There are no custom files.

4.9.12.2 Software Distribution Manager Context Diagram

The Software Distribution Manager, Tivoli Courier, does not interact with other system software. Tivoli Courier is installed on all of the platforms sending and/or receiving software distribution packages. Tivoli Courier on the source host platform communicates with Tivoli Courier installed on the receiving platforms.

4.9.12.3 Software Distribution Manager Architecture

The Software Distribution Manager COTS package, Tivoli Courier, is based on the Tivoli Management Platform, an architecture and set of fundamental COTS tools for managing client/server systems. Details of this architecture are provided in the Tivoli reference manuals.

4.9.12.4 Software Distribution Manager Process Descriptions

The Software Distribution Manager is a COTS product (Tivoli Courier) that handles adding, distributing, updating, and synchronizing all new software updates at a local site. Table 4.9.14.4-1 provides descriptions of the processes shown in the Software Distribution Manager architecture diagram in the Tivoli reference manuals.

Table 4.9.12.4-1. Software Distribution Manager Processes

Process	Type	COTS / Developed	Functionality
Software Distribution Manager {Tivoli Courier}	Other	COTS	• Provides a centralized software distribution capability to add new software, upgrade existing software with newer versions, and synchronize software on distributed systems. • Distributes software to multiple heterogeneous platforms concurrently. • Reports results of software distribution activity.

4.9.12.5 Software Distribution Manager Process Interface Descriptions

Not Applicable.

4.9.12.6 Software Distribution Manager Data Stores

The Software Distribution Manager's data stores consist of a database and a log file. Table 4.9.12.6-1 provides descriptions of the Tivoli Courier data stores.

Table 4.9.12.6-1. Software Distribution Manager Data Stores

Data Store	Type	Functionality
Tivoli Courier Database	Database	Tivoli Courier has a proprietary database management/database schema. Its database stores records concerning file package managers which consists of a set of file packages and a list of subscribing nodes and platforms for its file packages. A file package describes the source host location, the source file paths, the receiving platform path, specific actions performed on the files when they reach their destination, and log activity.
Log File	Other	The log file contains details about the results of software distribution activity, errors, and other process messages.

4.9.13 MLCI - Software License Manager Component

4.9.13.1 Software License Manager Functional Overview

The Software License Manager manages network licensing activities associated with using COTS products. The Software License Manager maintains information about license provisions,

meters use of installed licenses, and reports on licensing events and statistics for vendor software having embedded FLEXlm or iFOR/LS licensing technology.

Description: Software License Manager functionality is implemented using the COTS products FLEXlm and iFOR/LS along with ECS custom scripts and files. The custom scripts and/or files for the License Manager produce an ASCII log file monitored by MCI (Tivoli) to issue operations notifications. The log file contains all events the iFOR/LS server recorded in its database for the current day, except for events specifically excluded as specified in the License Manager configuration file.

Contents: The Software License Manager contains the following custom scripts and files:

- MsLiiFORLSMkDayLog (script) - makes license event records from an iFOR/LS database available in an ASCII log file for monitoring by MCI (Tivoli). This script extracts all current day records from the database, removes those containing patterns specified for exclusion, and copies the remainder to the log file. It is typically run as a scheduled job. Operating parameters are set via MsLiLicenseMgr.cfg, an associated configuration file.
- MsLiLicenseMgr.cfg (configuration file) - establishes the operating parameters for the License Manager custom script MsLiiFORLSMkDayLog, setting the following environment variables:
  - IFOR_LOGDIR - directory where iFOR/LS log file resides
  - IFOR_LOGFILE - name of file to contain iFOR/LS event records in ASCII form
  - IFOR_RPT_PGM - name of iFOR/LS report writer program
  - EXCLUDE_PATTERNS - patterns in records to be excluded from log
- README.FLEX (text file) - provides instructions for installing and configuring FLEXlm for the ECS.
- README.iFOR (text file) - provides instructions for installing and configuring iFOR/LS and its associated ECS customization files.

4.9.13.2 Software License Manager Context Diagram

Software License Manager runs at every ECS site, providing local network licensing services to requesting COTS applications and ECS operators. As shown Figure 4.9.13.2-1, it has a single interaction with another ECS COTS product, MCI's Tivoli, which notifies operators when "interesting" events occur. iFOR/LS maintains its log of events and errors in an internal database, but Tivoli can only monitor ASCII files. Consequently, Software License Manager processes Tivoli requests for an ASCII extract of the day's iFOR/LS events, storing the result in a file and returning request status to Tivoli.

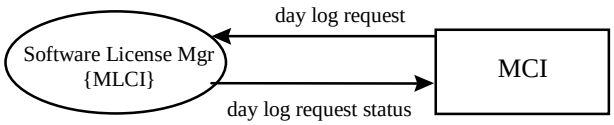

Figure 4.9.13.2-1. Software License Manager Context Diagram

4.9.13.3 Software License Manager Architecture

The Software License Manager, depicted in Figure 4.9.13.3-1, consists of two COTS products, FLEXlm and iFOR/LS, an ECS customization script for iFOR/LS, MsLiiFORLSMkDayLog, and related data files. Both COTS products use a client/server architecture with license servers responding to requests from client processes embedded in managed COTS applications or license manager utilities. Both permit multiple license servers to run concurrently, although iFOR/LS servers must run on separate hosts. Only iFOR/LS provides a graphical user interface.

FLEXlm consists primarily of license manager daemons, vendor daemons, license files, and client application code embedded in licensed application. Each FLEXlm server must have its own license file, and each server logs errors and licensing events to its own “debug file”. Options files are used to specify operating parameters for handling individual vendors’ products. Redundant FLEXlm servers can be configured to insulate against server failure; however, this requires three license server hosts.

iFOR/LS consists of the license server, code embedded in client applications, and the NCS location broker system. iFOR/LS stores licensing data as well as error and event logs in internal databases, and each server manages its own databases. (Licenses must be split among the iFOR/LS servers because iFOR/LS servers do not communicate or share licenses.) A user’s file can be configured to limit who can use product licenses. Products that can be run from only a single node rely on a nodelock file instead of the license server to determine if a license is available.

MsLiiFORLSMkDayLog is a custom script that updates log files with iFOR/LS events when invoked by MCI’s Tivoli product. Tivoli monitors these files and FLEXlm’s debug logs in order to notify operators when interesting events occur.

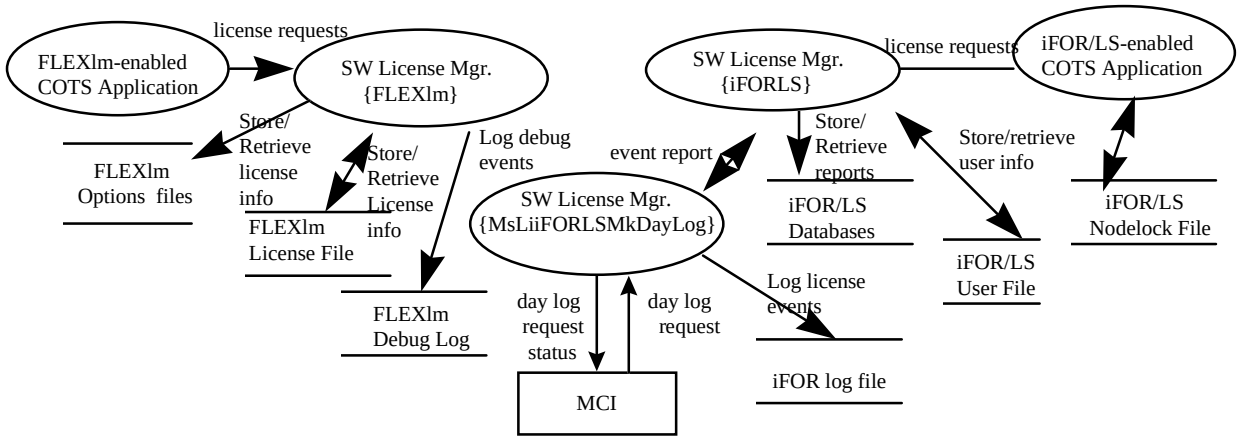

Figure 4.9.13.3-1. Software License Manager Architecture Diagram

4.9.13.4 Software License Manager Process Descriptions

Table 4.9.13.4-1 provides descriptions of the processes involved in software licenses management for a local site. It describes the processes within the COTS product and the custom code provided to track and monitor licenses on devices within the local network.

Table 4.9.13.4-1. Software License Manager Processes (1 of 2)

Process	Type	COTS / Developed	Functionality
Software License Mgr. (FLEXlm)	Other	COTS	The FLEXlm server daemon (lmgrd) with its associated command line utilities: • shuts down and restarts license daemons on a license server node and makes license data available to the servers. • manages license checkout and checkin processing for FLEXlm-enabled COTS products. • logs licensing events and errors to files on the local network. • removes a user's license for a specified feature. • displays the status of installed licenses and of network licensing activities; this includes listing licensed software features and their associated product versions, vendors, hosts, and expiration dates. • reports the hostid of a system (needed to obtain license key from vendors).

Table 4.9.13.4-1. Software License Manager Processes (2 of 2)

Process	Type	COTS / Developed	Functionality
Software License Mgr. (iFOR/LS)	Other	COTS	The iFOR/LS server daemon (netlsd) together with its associated command-line utilities: • shuts down and restarts license daemons on a license server node and make license data available to them. • installs licenses. • maintains a database of vendors and vendor products. • manages license checkout and checkin processing for iFOR/LS-enabled COTS products. • logs licensing events and errors to files on the local network. • generates reports of license management events. • removes a user's license for a specified feature. • displays the status of installed licenses and of network licensing activities; this includes listing licensed software features and their associated product versions, vendors, hosts, and expiration dates. • reports the hostid of a system (needed to obtain license key from vendors).
Software License Mgr (MsLiiFORLS MkDayLog)	script	Custom	Extracts event records from the iFOR/LS database and stores them in an ASCII log file in support of event monitoring for operator notifications.
FLEXlm-enabled COTS Application	Other	COTS	Client software within vendor products communicate with FLEXlm's license server and vendor daemons to request licenses for products users to run.
iFOR/LS-enabled COTS Application	Other	COTS	Client software within vendor products check for nodelock licenses in the nodelock file and communicate with the iFOR/LS license server to request concurrent use licenses for products users to run.

4.9.13.5 Software License Manager Process Interface Descriptions

The Software License Manager interacts with an MCI COTS product called Tivoli. Software License Manager processes "day log" requests for MCI Tivoli applications. On a scheduled basis, Tivoli runs the MsLiiFORLSMkDayLog script to have an extract of the day's iFOR/LS licensing events stored in an ASCII log file on the license server's host. Each extract contains all events the iFOR/LS server recorded in its database on the current day, except for those having patterns to exclude as specified in the License Manager Configuration File. The script exits with an appropriate status code Tivoli can monitor.

Internal interfaces also exist, generally between the license servers and their client applications embedded within COTS applications to handle license requests. These are not described here in detail as they are internal to the COTS.

Table 4.9.13.5-1 provides descriptions of the interface events shown in the Software License Manager architecture diagram.

Table 4.9.13.5-1. Software License Manager Interface Events

Event	Type	Interface Event Description
day log request	External	Periodic requests to Software License Managers to update the iFOR log file.
Day log request status	External	MsLiiFORLSMkDayLog returns an exit status indicating success or failure in processing the request.
Event report	Internal	Produces an ASCII log file containing records about the current day's iFOR/LS licensing events. Events are extracted from the iFOR/LS database using the iFOR/LS report utility and parameters defined in the MsLiLicenseMgr.cfg configuration file.
License requests	Internal	Communication among license servers and clients to establish connections and checkout, check-in, and monitor activity of licenses.
Store/Retrieve operating parameters	Internal	Store and retrieve parameters for operating individual vendor's products (i.e., temperature ranges, size thresholds, log capacity).
Store/Retrieve License info	Internal	Store and retrieve license information such as product name, model number, vendor, license number, license start/stop date, and license provisions.
Log debug events	Internal	The FLEXlm server logs all errors and license events (e.g., start/stop date, status of installed license, product version, host(s) to which servers are connected).
Log license events	Internal	Log a record of all license check in and check outtransactions, a history of license server activity.
Store/Retrieve reports	Internal	Stores and retrieves license information about the server, vendor, product, and licenses that users can request.
Store/Retrieve user information	Internal	Provides user information such as the purchaser of the license, when purchased, what host id the license is run on, license features and provisions, and expiration date of license.
Store/Retrieve licensed products	Internal	Files containing license provisions for one or more nodelock-license products. The files contain vendor ids, passwords, and annotations for each license.

4.9.13.6 Software License Manager Data Stores

License Manager's principal data stores are the FLEXlm license, debug, and option files; the iFOR/LS database, nodelock, and user files, and the iFOR day log file. FLEXlm files are described in the FLEXlm End User Manual. iFOR/LS files are described in the iFOR/LS Administrator's Guide. Table 4.9.13.6-1 provides descriptions of the data stores shown in the Software License Manager architecture diagram.

Table 4.9.13.6-1. Software License Manager Data Stores (1 of 2)

Data Store	Type	Functionality
FLEXlm license file	text file	A collection of records containing license provisions and passwords for one or more FLEXlm-enabled COTS products. They identify: • servers - name, hostid, and port number of the license manager daemon • daemons - name and path of vendor daemons that track licenses checked out and to whom. An options file can be named for each vendor daemon • features - description of the license to use a product. Each license server uses one license file, and operators combine license files received from vendors, as possible, to reduce the number of servers in the network. License files content and format is described in the FLEXlm End User Manual.
FLEXlm debug log	text file	A collection of records describing licensing errors and events that have occurred. Records contain a timestamp, an informational message, and the name of the daemon generating the message. Each license server (except redundant servers) writes to its own log file.
FLEXlm options files	text file	Collections of records that specify optional operating parameters for managing specific vendors' products. Options files are named in license files. There can be one options file for each vendor each license file specifies.
IFOR/LS databases	database	Three proprietary collections of records about iFOR/LS-enabled products, licenses, and server activity. Records are maintained by iFOR/LS COTS software and are located in files lic_db, cur_db, and log_file in directory /var/opt/ifor on the iFOR/LS license server host. File lic_db contains information about: • the server - Domain Name Services (DNS) name, socket information, target type, and target id • vendors - name, identifier, and password for each vendor whose product(s) managed by the server • products - name, version, product password, and license annotation for each vendor product managed by the server • licenses - password and provisions (including number and type, start/stop dates, timestamp, and annotation) for each license managed by the server File cur_db contains information about licenses available, licenses in use, wait queues, and users with licenses checked out. File log_file contains a history of iFOR/LS license server activity. Accessible only via the iFOR/LS report generation utility, it includes information about: • events - product, user, node, date/time, and description of license-related actions such as license grants and releases, users entering and exiting wait queues, and license status checks • errors - product, user, node, date/time, and description of errors detected by the license server • messages - product, user, node, date/time, and text of notifications logged by a software product or license server.

Table 4.9.13.6-1. Software License Manager Data Stores (2 of 2)

Data Store	Type	Functionality
iFOR/LS nodelock file	text file	One or more files containing the license provisions for one or more nodelocked-licensed products. The file resides on the machine where the licensed product runs and contains a vendor id, password, and annotation for each license.
iFOR/LS user file	text file	An optional collection of records that restricts who can use individual vendor products and that assigns priorities for users of products using wait queues.
iFOR day log file	text file	• A collection of records in ASCII format describing licensing events selected from those previously logged by iFOR/LS (See iFOR/LS databases: log_file above). The day log file, produced by script "MsLiiFORLSMkDayLog" according to parameters contained in its associated configuration file "MsLiLicenseManager.cfg", includes all events, errors, and messages iFOR/LS logged for the day up to the time the script is run, except for those containing patterns the configuration file specifies. The file is monitored by MCI (Tivoli) which issues operator notifications, as appropriate, based on type of event(s) described in the file.

4.9.13.7 Systems Management Subsystem Hardware Components

4.9.13.7.1MHCI Description

The MSS-MHCI include the following; two Application Servers, one MSS File Server, one CM (configuration management) server, two MSS Servers, one Tape Backup Server, and multiple PCs.

The Application Servers are SUN Server class machines. Detail specifications can be found per the site-specific, hardware design diagram, baseline document number 920-TDx-001. Because of their common configuration, these hosts can be configured interchangeably. Two MSS software CSCIs, MCI and MACI run on these hosts. Some of the key MCI functions are the MSS database management system and accountability management. As part of the MACI, custom and SNMP agents are configured to monitor and/or control managed objects distributed across heterogeneous platforms. Detailed mappings can be found per the site-specific hardware/software mapping, baseline document number 920-TDx-002.

A SUN SPARC Storage Array is dual ported between both hosts and provides storage for the MSS database management system and the IQ Report Writer tool. Detail configuration is specified per common disk partition, baseline document number 912-TDx-002.

The MSS File Server and CM Server are SUN Workstation class machines. Detail specifications can be found per the site–specific, hardware design diagram, baseline document number 920-TDx-001. Both servers are configured similarly with additional RAM allocated to the File Server due to file distribution loading. Two MSS software CSCIs, MLCI and MACI, run on these hosts. Some of the key MLCI functions are the Baseline Manager (XRP), Software Change Manager (ClearCase) and Change Request Manager (DDTS). As part of the MACI, custom and

SNMP agents are configured to monitor and/or control managed objects distributed across heterogeneous platforms. Detailed mappings can be found per the site-specific hardware/software mapping, baseline document number 920-TDx-002. Additional functionality provided by the File Server includes storage and processing of home directories, automounted COTS and distribution of custom code.

A SUN SPARC Storage Array is dual ported between both hosts and provides storage for the ClearCase VOBs and Views, DDTS, XRP, home directories, automounted COTS and distribution space. Detail configuration is specified per site specific disk partition, baseline document number 922-TDx-011.

The MSS Server and MSS Server Backup are HP High End Workstation class machines. Detail specifications can be found per the site–specific, hardware design diagram, baseline document number 920-TDx-001. Two MSS software CSCIs, MCI and MACI, run on these hosts. Some of the key MCI functions are network, enterprise, fault and performance management. These are all supported by a combination of Tivoli and HP OpenView COTS products. An additional key MCI component is trouble ticket (Remedy). As part of the MACI, custom and SNMP agents are configured to monitor and/or control managed objects distributed across heterogeneous platforms. Detailed mappings can be found per the site-specific hardware/software mapping, baseline document number 920-TDx-002.

A HP RAID device is dual ported between both hosts and provides storage for Remedy, HP OpenView, and Tivoli data. Detail configuration is specified per site specific disk partition, baseline document number 922-TDx-010.

The Tape Backup Server is a SUN Server class machine. Detail specifications can be found per the site–specific, hardware design diagram, baseline document number 920-TDx-001. This is a standalone host, which serves as the front end to a DLTL (Digital Linear Tape Library) used for global system DAAC backups. Two MSS software components run on this host and are the MCI and MACI. The key MCI functions are the network backup and restore components (Legatto Networker). As part of the MACI, Tivoli clients and SNMP agents are configured to monitor and/or control managed objects distributed across heterogeneous platforms. Detailed mappings can be found per the site-specific hardware/software mapping, baseline document number 920-TDx-002.

A DLTL with more then 1 TB of capacity is directly attached to the Tape Backup Server. Via the Legatto Networker Server, system data copies and restores are performed with the Tape Backup Server functioning as the intermediate between the Legatto clients and the DLT. Detail configuration is specified per site specific disk partition, baseline document number 922-TDx-010.

Multiple Pentium PCs are used at each DAAC site in support ofoffice automation requirements. Detail specifications can be found per the site–specific, hardware design diagram, baseline document number 920-TDx-001. These are standalone hosts, which enable operators to perform policy and procedure management. One MSS software component runs on this host and is the MCI. Detailed mappings can be found per the site-specific hardware/software mapping, baseline document number 920-TDx-002.

In general, custom code and applications are loaded on the internal disks of all hosts. This prevents dependencies on specific hosts or any peripherals. For cost efficiency, selective application servers are stored in a RAID and accessed by one host at any time.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.10 Internetworking Subsystem (ISS) Overview

The Release 4 Internetworking Subsystem (ISS) contains one hardware configuration item (HWCI), the Internetworking HWCI. INCI provides internetworking services based on protocols and standards corresponding to the lower four layers of the OSI reference model as described below.

Transport Protocols

ECS provides IP-based connection-oriented and connectionless transport services. The connection-oriented service is implemented using TCP, while UDP is used for connectionless transport. Higher layer applications use one or the other based on such requirements as performance and reliability.

Transmission Control Protocol (TCP), specified in RFC 793 (as of 08//98), is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols to support multi-network applications. It provides for reliable inter-process communication between pairs of processes in host computers attached to networks within and outside ECS. Because TCP assumes it may obtain potentially unreliable datagram service from the lower level protocols, it involves additional overhead due to the implementation of re-transmission and acknowledgment processes.

The User Datagram Protocol (UDP), specified in RFC 768 (as of 08//98), provides a procedure for application programs to send messages to other programs with minimal overhead. The protocol is transaction oriented and delivery of data is not guaranteed, since there is no acknowledgment process or re-transmission mechanism. Therefore, applications requiring ordered and reliable delivery of data would use TCP.

Network Layer Protocols

The network layer provides the functional and procedural means to transparently exchange network data units between transport entities over network connections, both for connection-mode and connectionless-mode communications. It relieves the transport layer from concern of all routing and relay operations associated with network connections.

The Internet protocol (IP) Version 4, specified in RFC 791 (as of 08//98), is the ECS supported network protocol, based on its dominance in industry usage and wide community support. As part of IP support, ICMP and ARP are also supported.

Physical/Datalink Protocols

Physical and data-link protocols describe the procedural and functional means of accessing a particular network topology. For the Release 4 DAAC and SMC networks, the data-link/physical protocols to be implemented are Fiber Distributed Data Interface (FDDI) and Ethernet. (FDDI is a 100Mbps token-passing network topology, and Ethernet is a 10 Mbps bus topology.)

High-Performance Parallel Interface (HIPPI) networks form part of the networks at some DAACs (GSFC, LaRC, and EDC) to handle the high data volumes between the Processing and

Data Server subsystems. The HIPPI implementation involves running IP-over-HIPPI. (Large TCP window sizes are used in order to achieve high throughput rates on the HIPPI networks.)

Other technologies such as Gigabit Ethernet and Asynchronous Transfer Mode (ATM) are being considered for insertion in ECS DAAC networks.

Internetworking Hardware HWCI (INCI)

This HWCI provides the networking hardware for internal and external DAAC, SMC, and EOC connectivity. The HWCI includes FDDI switches, concentrators and cabling; Ethernet hubs and cabling; routers and cabling; HIPPI switches and cabling; and network test equipment. Each network hardware device is discussed in detail in Section 4.10.2

4.10.1 Internetworking Subsystem Description

4.10.1.1 DAAC LAN Architecture

This section provides an overview of the Release 4 DAAC network architecture. Information on DAAC specific implementation level detailed designs can be found in Section 4.10.1.5.

The generic architecture for Release 4 DAAC Local Area Networks (LANs) is illustrated in Figure 4.10.1.1-1. The topology consists of a User Network (FDDI at all sites), a Production Network (FDDI at all sites), and a HIPPI Network (for processing to data server flows at GSFC, EDC, and LaRC). The creation of separate User and Processing networks allows processing flows to be unaffected by user pull demands, and the introduction of the high-speed HIPPI Network provides adequate bandwidth to the Processing and Data Server subsystems' need to transfer large volumes of data. Each of the networks is discussed in more detail below.

Note that not all sites have the complete complement of hardware and subsystems shown in Figure 4.10.1.1-1. For instance NSIDC does not have a HIPPI network because HIPPI is not needed to satisfy the relatively moderate processing flows.

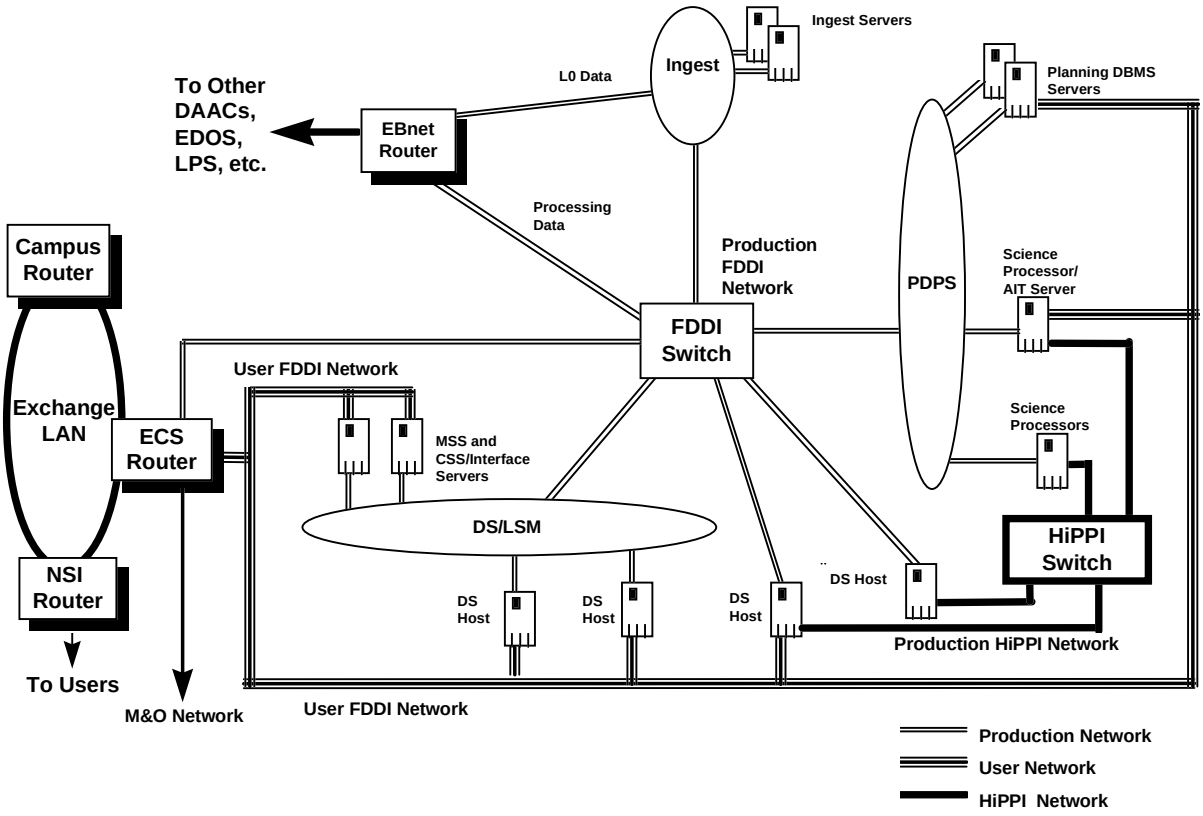

Figure 4.10.1.1-1. Release 4 DAAC Networks: Generic Architecture Diagram

The Production Network consists of multiple FDDI rings supporting the DAAC subsystems and connections to external production systems (such as EDOS and other ECS DAACs) via EBnet. At GSFC, EDC, and LaRC some Data Server hosts are contained on a dedicated FDDI ring in order to provide adequate bandwidth for DAAC-to-DAAC processing flow requirements. A dedicated FDDI ring provides access to the EBnet router to handle the DAAC-to-DAAC production flows. The FDDI Switch discussed in Section 4.10.2.1 is the central device connecting the FDDI rings together, and it provides the necessary routing and filtering control.

The User Network is an FDDI-based LAN connecting the users (via NSI, local campuses, general Internet, etc.) to the DAAC hosts responsible for providing user access. It has the main advantage of separating user and production flows. This allows DAAC processing data flows to be unaffected by user demand, so that even unanticipated user pulls do not hinder the Production Network. Users do not have access to any other hosts, such as Ingest or Data Processing devices. CSS and MSS servers are connected to the User Network but do not allow direct user access. These connections are required for communications with outside networks for such things as name lookups and receipt of Internet mail, as well as communication with and monitoring of the DAAC's interfaces to the user community (such as NSI and the local campus). The User

Network connects to NSI and the local DAAC campuses through an ECS router (discussed in Section 4.10.2.2) which provides the necessary routing and filtering controls.

The individual FDDI rings for both the User and Production Networks are implemented using FDDI concentrators to provide ease of wiring and central points of management. All DAAC hosts have FDDI interfaces and are attached directly to the FDDI rings. Workstations have single-attached FDDI cards, whereas the high-performance servers and processors on the Production Network have dual-attached FDDI cards to provide redundancy. The interfaces of these machines that are also on the User Network have single-attached interface cards. Dual-attached hosts are dual-homed to two separate FDDI concentrators. Printers, PCs, and x-terminals are connected to a FDDI ring via an FDDI-to-Ethernet hub.

The HIPPI Network interconnects Data Server hosts/devices and Science Processors in order to provide a high-speed network to handle the large data transfers between the two subsystems. The HIPPI network is implemented via a central HIPPI switch with switched interface ports, of at least 800 Mbps, connected directly to the high-powered processing and storage hosts. The HIPPI Network shifts the numerous transfers of large volumes of data onto a dedicated high-speed fabric.

4.10.1.2 SMC Network Architecture

The SMC network architecture, as illustrated in Figure 4.10.1.2-1, consists of two FDDI LANs connected to the GSFC DAAC ECS router. MSS and CSS servers are connected to one of the FDDI rings, and PC workstations and a printer are attached to an Ethernet network bridged to the FDDI ring via an Ethernet-to-FDDI hub. The Bulletin Board Server (BBS) and two FTP servers are attached to the second FDDI ring.

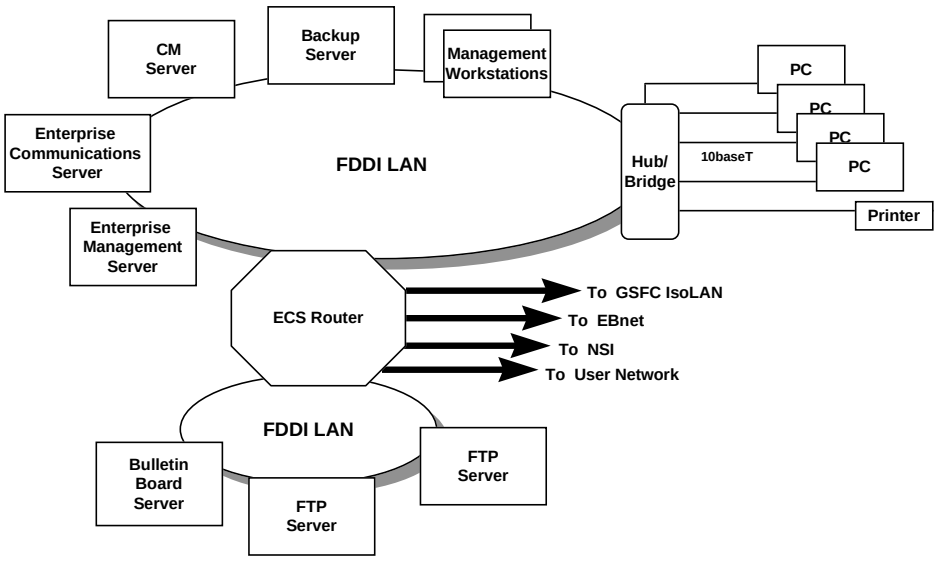

Figure 4.10.1.2-1. SMC Network Architecture Diagram

4.10.1.3 DAAC Addressing and Routing Architecture

The Planning and Data Processing, MSS, CSS, and Data Server/Data Management subsystems (collectively known as the Production Network) are connected to the FDDI switch on switched ports. They are assigned a Class C address space. The Ingest subsystem and the EBnet router are connected to the ECS FDDI switch on routed ports. They are assigned Class C or subnetted Class C address space. User Networks are connected to the ECS Router on routed ports. They are assigned Class C addresses. At GSFC, a subnet of the User Network Class C address space is used for the SMC. A subset of hosts (belonging to the Data Server and Processing subsystems) that are connected to the FDDI Production Network, also have interfaces connected to a HIPPI switch forming a HIPPI production network. The hosts are assigned private addresses as specified in RFC 1597 (as of 08//98). Documents that list IP address assignments to all hosts and network attached devices are listed in Table 4.10.1.5-1. All ECS address space (except for addresses used on HIPPI networks) is provided by EBnet from Class C address blocks designated by NSI.

Routing Information Protocol (RIP) is the protocol used to route IP packets within ECS as well as to/from external networks. ECS Production Networks are advertised to all ECS via EBnet. ECS ingest networks are advertised to data providers such as EDOS via EBnet. User Networks are advertised via RIP to NSI and campus networks.

4.10.1.4 Network-based Security Architecture

The Release 4 network architecture provides basic levels of security to isolate and protect hosts and subsystems within the DAACs and SMC. Note that in addition to network-based security; ECS has implemented other security measures, such as DCE-based authentication and authorization, Kerberized telnet and FTP secure shell (SSH), and DCE access control lists (ACLs) which are discussed in CSS sections of this document.

At each ECS router connecting to external EOSDIS networks (such as EBnet) and external user networks (such as NSI), security filters have been implemented to control access to DAACs. These network and transport-layer filters control types of traffic that pass through the FDDI switch or ECS router, and they are able to control access to individual hosts as well as to whole subsystems.

4.10.1.5 Internetworking Subsystem Detailed Design

The ISS implementation level detailed design is documented in the documents listed in Table 4.10.1.5-1. All of the documents are under configuration control and can be obtained from ECS Configuration Management. The same information shown in Table 4.10.1.5-1 can be found at the WWW page http://cmdm.east.hitc.com/baseline. The documents are not on line for security reasons. Therefore special authorization is needed for their release.

Table 4.10.1.5-1. Internetworking Subsystem Baseline Documentation List

Document Name	EDC	GSFC	LaRC	NSIDC	SMC
Hardware/Network Diagram	921-TDE-002	921-TDG-002	921-TDL-002	921-TDN-002	921-TDS-002
Host IP Address Assignment Table	921-TDE-003	921-TDG-003	921-TDL-003	921-TDN-003	921-TDS-003
Network Hardware IP Address Assignment	921-TDE-004	921-TDG-004	921-TDL-004	921-TDN-004	921-TDS-004
Dual - Homed Host Static Routes	921-TDE-005	921-TDG-005	921-TDL-005	921-TDN-005	N/A
Ingest Host Static Routes	921-TDE-006	921-TDG-006	921-TDL-006	N/A	N/A

4.10.2 Release 4 Network COTS Hardware

The Release 4 DAAC and SMC LANs contain six types of COTS hardware: FDDI switches, Routers, Remote access servers/modems, FDDI concentrators, Ethernet hubs, and HIPPI switches. As described above, the FDDI rings within the DAACs are implemented via FDDI concentrators, and the FDDI switch is used to connect multiple Production Network FDDI rings together (refer to Figure 4.10.1.1-1). The FDDI-to-Ethernet hubs are used to connect PCs, printers, x-terminals, and remote access servers in the DAACs. At the SMC, the FDDI-to-Ethernet hubs are used to connect printers, x-terminals, and PC workstations. The Routers are used to provide access to external networks (NSI and Campus nets) via the User Network. The Remote access servers and modems provide access for instrument teams that want dial-up access at some of the DAACs. The HIPPI switches connect the Data Server and processing hosts with a high-speed fabric to be used for transferring large volumes of data between the two subsystems (Data Server and Data Processing). Table 4.10.2-1 provides a list of networking hardware used in ECS Release 4 networks.

The following descriptions of Release 4 Network Hardware devices are provided as illustrative detail. All details of the hardware configuration should be verified with the appropriate Hardware/Network Diagram shown in Table 4.10.1.5-1.

Table 4.10.2-1. Release 4 Networking Hardware for ECS Networks (1 of 2)

Networking Hardware	Vendor
FDDI Switch	FORE PowerHUB 8000
Router (ECS Router)	Cisco 7507 (7513 at GSFC)
HIPPI Switch	Essential Communications EPS-16
Remote Access Server/Modems	Cisco 2509/Hayes OPTIMA 288, V.34.
FDDI Concentrator	Bay Networks 2914-04 concentrator with 12 M & 1 A/B port
Ethernet Hub	Cabletron MicroMMAC-22E; used for PCs, remote access sensors, printers and x-terminals
FDDI Cables	Multimode fiber cables with MIC connectors

Table 4.10.2-1. Release 4 Networking Hardware for ECS Networks (2 of 2)

Networking Hardware	Vendor
Ethernet Cables	10baseT connection to printers, PCs, x-terms, printers and remote access servers
HiPPi Cables	Multi-wire copper cable for parallel HIPPI interface

4.10.2.1 ECS FDDI Switch

The ECS FDDI switch is the FORE PowerHUB 8000 with FDDI interface modules (DAS interfaces) and a powerful packet engine. The switch forms the core of the ECS Production network by interconnecting all FDDI segments that form the ECS production network as well as ingest segments. It also interfaces with EBnet. At the EOC, the FDDI switch interconnects the production, support and EOC M&O networks. All ports on the switch can be configured to switch or route giving the flexibility needed for configuring interfaces for data link layer or network layer connectivity.

The switch has redundant power supply and fan units. All interface modules are hot swappable.

4.10.2.2 ECS Router

The ECS Router is a Cisco 7500 series router (7513 at GSFC and 7507 at all other DAACs) running Cisco's Internetwork Operating System (IOS). All routers have Versatile Interface Processor (VIP) boards populated with FDDI DAS ports. The ECS Router is a key item of ECS DAAC networks in that it provides connectivity to the Internet via its interface with NSI. All ECS User Networks and M&O Networks at each DAAC are connected to the ECS Router. In addition, the SMC network is connected to the GSFC ECS Router.

The ECS Router has redundant power supply and fan units. All interface modules are hot swappable.

4.10.2.3 HIPPI Switch

The ECS HIPPI switch is part of ECS DAAC networks at GSFC, EDC, and LaRC is an Essential Communications EPS-16 switch capable of supporting up to 16 parallel or serial HIPPI interface modules. It also has an Ethernet port for switch management. The HIPPI switch forms the core of the HIPPI fabric interconnecting Data Server and processing hosts providing capacity of up to 800 Mbps per connection.

4.10.2.4 Remote Access Server and Modems

The ECS Remote Access Server (implemented at EDC) is a Cisco 2509 access server with 1 Ethernet and 8 asynchronous ports. It provides dial up access to instrument team members that need such service. Two Hayes OPTIMA 288, V.34 modems are attached to two asynchronous ports. The Ethernet port is used for connectivity to an Ethernet hub on the ECS User network.

4.10.2.5 Ethernet Hub

The ECS Ethernet hub (10BaseT) is a Cabletron MicroMAC-22E (with BRIM-F6 module) Ethernet-to-FDDI hub. It is a stackable hub with 1 A/B FDDI port and comes with 12 or 24 shared Ethernet ports. All ECS printers, x-terminals, PC workstations and remote access servers are connected to Ethernet hub.

4.10.2.6 FDDI Concentrator

The ECS FDDI Concentrator is a Bay Networks System 2000 Model 2914-04. It is a stackable concentrator with 12 M Ports and 1 A/B Port (all MIC interfaces). All FDDI rings with multiple nodes on them are formed using several concentrators interconnected to form a ring. Ethernet hubs with FDDI uplinks are also connected to DAAC FDDI networks via the FDDI concentrators.

This page intentionally left blank.

5. Limitations of Current Implementation

DATA SERVER SUBSYSTEM

Science Data Server (SDSRV) CSCI

- **Operator GUI**: There is no support for re-installation of ESDTs from the GUI. Reinstallation of ESDTs is supported through a command line Unix Shell script interface. Corresponding data must be manually deleted from the Subscription Server, Advertising and Data Dictionary.
- Metadata update services only support QA metadata.
- There is no persistence of Client requests. Requests must be re-submitted.
- There is no support for Access Control List checking.
- The spatial search capability is limited to returning only granules of the same basic shape as submitted in the request. For example, a search request for all granules intersecting a certain Bounding Rectangle, the SDSRV only returns granules of type Bounding Rectangle. Granules of other spatial shapes - Gpolygons, Circles, Points are not returned.
- Spatial shapes with an internal arc greater than 180 degrees are interpreted as the complement of the given shape. This applies both to granules and client search areas. This is due to a COTS implementation.
- Certain queries containing constraints against a plural spatial data type (Gpolygon for example) and another constraints against a multi-value attribute (Additional Attributes for example) cannot be performed due to an implementation limitation within a COTS product.

CLIENT SUBSYSTEM

Workbench Software CSCI

- The DAR Tool requires the use of DCE to communicate to the DAR Communications Gateway. As a result, users must have DCE installed on their system. Their workstation must be configured in the DCE cell in order for it to communicate with the DAR Communications Gateway.
- The Desktop does not allow users to create icons through its GUI interface. Users would have to know how to change the Desktop files in order to add new application or document objects to the Desktop.

PLANNING SUBSYSTEM

Production Planning CSCI

- The Production Request Editor has not been optimized when creating Production Requests. The creation of large production requests with many input and output granules that may take some time. This is mostly due to multiple database accesses; some database accesses may need to be replaced in the future with stored procedures to improve performance.
- The Production Planning Workbench has not been optimized when creating Production Plans. The creation of plans with many data processing requests may take some time. In addition to the database accesses that slow down the Production Request Editor, the Production Planning Workbench's scheduling algorithm may need to be optimized.
- There is no inter-DAAC planning at this time.

INFRASTRUCTURE SUBSYSTEM

ASTER DAR Gateway CSCI

- The ASTER DAR Client or end users must have valid DCE login in order to communicate with ASTER DAR Gateway.
- The user must be authorized to perform any of seven ASTER DAR Gateway functions including ModifyDAR, getxARStatus, getSubxARStatus, getxARContents, querytxARSummery, and queryxARScenes.
- The gateway itself does not extend the DAR functionality; but is limited by the functionality provided by the ASTER GDS through the API set.

E-mail Parser Gateway CSC

- E-mail Parser Gateway only handles Expedited Data Request.
- E-mail Parser Gateway only handles FTPPUSH for media type.

Landsat7 Gateway CSC

- The authentication is only limited to comparing the login name and password with the ones stored in the configuration file.
- Landsat7 Gateway does not have a restriction on the number of threads that are spawned, meaning in theory there could be too many threads running at any given time, but in practice this is unlikely to happen.
- Landsat7 Gateway doesn't provide any queuing mechanism. It doesn't have any intelligence. Its functionality is to PASS information between landsat7 system and the ECS ingest.

Subscription Server CSC/Subscription GUI

- The user can only subscribe to future granules.
- The Subscription Server validates only String type qualifiers.
- There is no persistence for trigger request, subscription request, event registration request, event and subscription update and delete. These requests may be lost or partially finished if some system (including DCE) problems happen when they are processed.
- Support for trigger persistence has not been merged into the current baseline.
- E-mail contents are not clear.
- Subscription Server does not check security issues for updating its database tables. It assumes the client application does this.
- The Operator GUI cannot register or update events on the server side.
- It takes a long time to bring up the Operator GUI.
- The Operator GUI on-line help has not been completed.

MSS SUBSYSTEM

MCI CSCI - Security CSC

- If an unauthorized user gains access to a host despite security measures, it is not detected until the next detection interval expires, since this is only checked periodically. (If the detection intervals were decreased, the system uses too much of its processing power for monitoring itself.)
- The security implementation requires the operator to perform security tasks such as running SATAN and running Crack manually.
- The configuration files for TCP Wrappers can become difficult to manage when multiple versions exist as they surely do. If administrators setup a “back door,” the system security can be more easily compromised.

MCI CSCI - Accountability CSC

- There is no retry in place for Database updates or inserts and errors are logged as low priority. Currently, Accountability does not attempt to reconnect to the Database once the connection is lost. Accountability must be restarted to re-establish the connection.

MCI CSCI - Trouble Ticketing CSC

- Trouble Tickets that are forwarded from a DAAC to the SMC are set to a forwarded state in the DAAC. A manual process is necessary to receive notification that the Trouble Ticket has been closed at the SMC and will now be closed at the DAAC.
- While the Trouble Ticket is being worked at the SMC, the M&O operator at the DAAC has little insight as to the current status of the Trouble Ticket.

- For an overall status of Trouble Tickets in the ECS system, reports must be run at each DAAC and forwarded via e-mail where they can be consolidated.

MCI CSCI - Network and Enterprise Management CSC

- If the HP OpenView processes go down, all network hardware and custom software monitoring is incapacitated.
- If the TMR server goes down, all host and COTS software monitoring is incapacitated.
- A single log file adapter monitors all log files and the entry is passed sequentially through all filters that are configured on a particular host. Thus, a generic string such as ERROR would not be of any use in multiple log file configurations. The event would be generated using the first matching configuration and possibly be reported as an event occurring with the incorrect log file.
- Due to an unacknowledged bug in the Tivoli log file adapters, running these adapters causes the syslog to crash. Until Tivoli resolves this problem, Tivoli log file adapters are going to be turned off (per an engineering directive) and there is going to be no Tivoli monitoring of COTS logs.

MCI CSCI - Network and Management CSC

- Network and Management component: ESSM is no longer being supported by Platinum Technologies. ECS will need to migrate to a new product by Platinum called DBVISION. This product has similar functionality to ESSM and is provided as part of Drop 5A.

MLCI CSCI - Baseline Manager CSC

- Control item identifiers - XRP-II uses centralized database technology and is separately installed at each ECS site. This necessitates a special scheme for assigning identifiers to control items so that site may safely exchange database records. For example, the SMC must be able to distribute centrally maintained release records to multiple sites without interfering with records the sites locally maintain there. Similarly, the SMC must be able to absorb copies of site-maintained records to form the consolidated picture of system-wide baselines without contaminating centrally maintained data. To distinguish between centrally maintained and site-maintained records, Baseline Manager expects that identifiers of site-maintained control items have a site 3-character prefix.
- Data entry screens offer form and table views for browsing and editing data records. Table view driver programs cannot handle the number and size of fields used in the form view of numerous screens. Where limitations exist, fields that appear in table view were chosen either because they are best suited to identifying and classifying control items or because they are likely to be used in multi-record operations.
- Import file directories - XRP-II uses the name contained in the IMPORTPATH environment variable as a destination when exporting data records to other sites using the FTP service. Consequently, the directory used to receive the data should have the same name at each site.